

DEFINIÇÃO E CONSTRUÇÃO DE AMBIENTES DE DESENVOLVIMENTO DE
SOFTWARE ORIENTADOS À ORGANIZAÇÃO

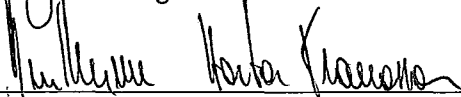
Karina Villela de Carvalho Lima

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS
DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO
DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A
OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE
SISTEMAS E COMPUTAÇÃO.

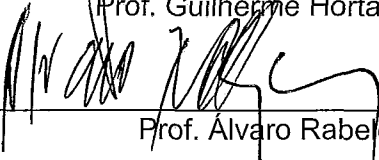
Aprovada por:



Prof. Ana Regina Cavalcanti da Rocha, D.Sc.



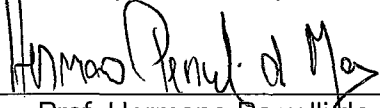
Prof. Guilherme Horta Travassos, D.Sc.



Prof. Alvaro Rabelo Alves Jr., L.D.



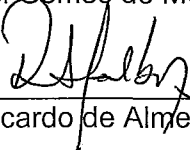
Prof. Claudia Maria Lima Werner, D.Sc.



Prof. Hermano Perrelli de Moura, Ph.D.



Prof. Manoel Gomes de Mendonça Neto, Ph.D.



Prof. Ricardo de Almeida Falbo, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2004

LIMA, KARINA VILLELA DE CARVALHO

Definição e Construção de Ambientes de
Desenvolvimento de Software Orientados a
Organização [Rio de Janeiro] 2004

VIII, 329 p. 29,7 cm (COPPE/UFRJ, D.Sc.,
Engenharia de Sistemas e Computação, 2004)

Tese - Universidade Federal do Rio de
Janeiro, COPPE

1. Ambientes de Desenvolvimento de Software
2. Gerência de Conhecimento
3. Ontologia

I. COPPE/UFRJ II. Título (série)

Agradecimentos

A Ana Regina, pela oportunidade de realizar este trabalho, pela orientação sempre oportuna e adequada, e por me hospedar várias vezes em sua casa.

A Guilherme, pela orientação, mas, principalmente, por ter sido fonte constante de estímulo e auxílio.

A Dr. Álvaro, pela oportunidade de aplicar as idéias desta tese na Fundação Bahiana de Cardiologia e pelo empréstimo de um computador no momento em que muito precisei.

A Ricardo Falbo, por responder sempre amável e prontamente aos inúmeros e-mails com dúvidas ou com partes da ontologia de organização para avaliar.

A todos os professores da banca, pela revisão cuidadosa da tese e vários comentários que contribuíram para a obtenção desta versão final.

A Gleison, Mariano, Sávio e Sômulo, a super-equipe Taba, por estarem sempre disponíveis para ajudar, atender as minhas solicitações e tirar as minhas dúvidas.

A Kathinha, AnaCog, Lu, Dani Onnis, Mariella e Fred, que doaram parte de seu tempo para colaborar com alguma atividade. A Kathinha, em especial, pelos valiosos conselhos.

A Flavinha, por ter me recebido em sua casa em um momento muito especial.

Aos vários amigos, que enviaram mensagens de apoio e confiança, todos na torcida para que tudo desse certo.

A meu pai e minha mãe, pela preocupação e carinho eternos, como também pelo apoio logístico fundamental em vários momentos.

A minha família, pela preocupação, e a Sandro e Sandrinha, em especial, pelas várias acolhidas no Rio de Janeiro.

A Marius, pelo carinho, compreensão e apoio, pela eterna espera e pelas saudáveis fugidas do mundo.

A Dani, minha florzinha, por ter agüentado firme todo este período de várias privações e pela inversão de papéis, cuidando de mim ao longo dos últimos meses.

A Deus, por me fazer seguir sempre em frente.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

DEFINIÇÃO E CONSTRUÇÃO DE AMBIENTES DE DESENVOLVIMENTO DE SOFTWARE ORIENTADOS A ORGANIZAÇÃO

Karina Villela de Carvalho Lima

Março/2004

Orientadores: Ana Regina Cavalcanti da Rocha
Guilherme Horta Travassos

Programa: Engenharia de Sistemas e Computação

Este trabalho define Ambientes de Desenvolvimento de Software Orientados a Organização (ADSOrg) como sendo ambientes de desenvolvimento de software que apóiam a gerência do conhecimento relevante para os processos de desenvolvimento e manutenção de software de suas respectivas organizações. Além disso, propõe um modelo que especifica os componentes necessários para que tais ambientes contemplem os seus requisitos. Uma pesquisa do tipo enquete foi conduzida com a finalidade de orientar futuros esforços de incorporação de conteúdo e desenvolvimento de serviços e ferramentas para ADSOrg. Por fim, uma estratégia para a construção de ADSOrg a partir de um meta-ambiente foi definida, implementada e utilizada, evidenciando o apoio fornecido pelas ferramentas desenvolvidas.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

ENTERPRISE-ORIENTED SOFTWARE DEVELOPMENT ENVIRONMENTS

Karina Villela de Carvalho Lima

March/2004

Advisors: Ana Regina Cavalcanti da Rocha

Guilherme Horta Travassos

Department: Computer Science and System Engineering

This thesis defines Enterprise-Oriented Software Development Environments (EOSDE) as software development environments that support the management of relevant knowledge for organizational software development and maintenance processes. Furthermore, a model is presented that specifies the necessary components to satisfy the requirements of such environments. A survey was carried out in an attempt to guide future work on incorporating content and developing services and tools for EOSDE. Finally, a strategy for the construction of EOSDE from a meta-environment was defined, implemented and used demonstrating the support provided by the tools developed.

Capítulo 1 – Introdução.....	1
1.1 Engenharia de Software.....	1
1.2 Gerência do Conhecimento.....	2
1.3 Ambiente de Desenvolvimento de Software Orientado à Organização.....	4
1.4 Histórico de Trabalho.....	5
1.5 Organização da Tese.....	9
Capítulo 2 - Gerência do Conhecimento em Engenharia de Software.....	10
2.1 Introdução.....	10
2.2 Conceito de Conhecimento.....	10
2.3 Tipos de Conhecimento.....	11
2.4 Processo de Conversão e Evolução do Conhecimento.....	11
2.5 Conceito de Gerência do Conhecimento.....	12
2.6 Frameworks de Gerência do Conhecimento.....	13
2.7 Fábrica de Experiência: o mais popular dos frameworks.....	16
2.8 Processo de Implementação de Gerência do Conhecimento.....	17
2.9 Atividades de Gerência do Conhecimento.....	19
2.10 Infra-estrutura Técnica de Gerência do Conhecimento.....	21
2.11 Sistemas.....	32
2.12 Fatores Não Técnicos.....	39
2.13 Lições Aprendidas, Recomendações e Questões.....	41
2.14 Conclusão.....	42
Capítulo 3 - Gerência do Conhecimento Orientada a Processos.....	43
3.1 Introdução.....	43
3.2 Integração de Gerência do Conhecimento com Modelagem de Processos.....	43
3.3 Modelagem de Processos.....	44
3.4 Justificativas e Requisitos.....	44
3.5 Abordagens para Integração de Gerência do Conhecimento com Modelagem de Processo.....	45
3.6 Sistemas e Relatos de Experiência.....	46
3.7 Conclusão.....	56

Capítulo 4 - Ambientes de Desenvolvimento de Software.....	57
4.1 Introdução.....	57
4.2 Histórico.....	57
4.3 ADS Centrados em Processo.....	59
4.4 ADS Orientados a Domínio.....	63
4.5 Projeto TABA.....	66
4.6 Projeto MILOS.....	75
4.7 Projeto ODE.....	77
4.8 Outros Projetos.....	79
4.9 Conclusão.....	81
Capítulo 5 - Ambientes de Desenvolvimento de Software Orientados à Organização.....	82
5.1 Introdução.....	82
5.2 Conceito de ADSOrg.....	82
5.3 Tipos de ADSOrg.....	83
5.4 Requisitos dos ADSOrg.....	84
5.5 Modelo de ADSOrg.....	86
5.6 Modelagem de Processos em ADSOrg.....	92
5.7 Ontologia de Organização.....	103
5.8 Conclusão.....	134
Capítulo 6 – Pesquisa sobre a Importância dos Tipos de Conhecimento.....	135
6.1 Introdução.....	135
6.2 Definição da Pesquisa.....	135
6.3 Planejamento da Pesquisa.....	136
6.4 Operação da Pesquisa.....	141
6.5 Análise dos Dados da Pesquisa.....	142
6.6 Interpretação dos Resultados da Pesquisa.....	149
6.7 Discussão sobre o Resultado da Pesquisa.....	155
6.8 Conclusão.....	157

Capítulo 7 – Construção de Ambientes Orientados a Organização na Estação TABA	158
7.1 Introdução	158
7.2 Estratégia para Construção de ADSOrg na Estação TABA	158
7.3 Funções e Ambientes da Estação TABA	159
7.4 Requisitos dos Ambientes da Estação TABA	160
7.5 Processos para Adaptação de Ambientes na Estação TABA	166
7.6 Arquitetura da Estação TABA	171
7.7 Considerações Gerais de Implementação	175
7.8 Ferramentas Desenvolvidas	176
7.9 Estado Atual da Estação TABA	180
7.10 Cordis-FBC: um Ambiente Configurado TABA	183
7.11 Conclusão	203
Capítulo 8 – Conclusão e Perspectivas Futuras	204
8.1. Conclusão	204
8.2. Perspectivas Futuras	211
Referências Bibliográficas	213
Anexos	231
Anexo I – Instrumentação da Pesquisa: Questionários	231
Anexo II – Instrumentação da Pesquisa: Sistema para Aplicação de Questionários na Web	242
Anexo III – Valores do Teste de Hipóteses e Análise Descritiva dos Dados	246
Anexo IV – Classificação dos Tipos de Conhecimento quanto à Importância	256
Anexo V – Processo de Configuração do Meta-ambiente TABA	280
Anexo VI – Atividades de Preparação do Ambiente Configurado	293
Anexo VII – Visão Parcial do Modelo de Classes da Estação TABA	297
Anexo VIII – Descrição das Ferramentas Config e ProcKnow	303
Anexo IX – Modelos de Classes da Ferramenta ProcKnow	326

Capítulo 1 - Introdução

A Engenharia de Software lida de forma intensa com conhecimento. O processo de transformação de uma solicitação abstrata do cliente em declarações de linguagem de programação requer e produz vários conhecimentos. Estes conhecimentos precisam ser organizados, armazenados e disponibilizados, de forma que novo conhecimento possa ser construído a partir de conhecimentos já existentes e, conseqüentemente, as solicitações dos clientes possam ser rápida e adequadamente atendidas. Verifica-se, desta forma, a importância da administração sistemática dos conhecimentos relevantes neste contexto.

No entanto, é preciso responder a uma questão: como integrar esta administração sistemática do conhecimento com as atividades de Engenharia de Software propriamente ditas?

O objetivo desta introdução é descrever o problema abordado nesta tese, apresentar as abordagens que fundamentam a solução proposta e brevemente relatar o caminho percorrido para construí-la.

1.1 Engenharia de *Software*

DESOUZA [1] destaca que a Engenharia de *Software* é um domínio de conhecimento amplo, composto de vários subdomínios, onde raramente um indivíduo tem todo o conhecimento necessário a um projeto. Além disso, em um projeto de *software*, são requeridos conhecimentos sobre computação, sobre o domínio da aplicação e sobre a aplicação propriamente dita [2].

LINDVALL *et al.* [3] destacam a natureza não repetitiva, experimental e evolutiva da Engenharia de *Software*. Uma vez que cada projeto define uma situação específica e não existem soluções adequadas a todas as situações, é necessário descobrir quais são as abordagens que apresentam melhores resultados de acordo com a situação apresentada [4] e, quando necessário, evoluir as abordagens existentes para atender às necessidades de novas situações. Em termos de novas situações, avanços tecnológicos extremamente rápidos e a constante ampliação do papel desempenhado pelo *software* na vida atual têm direcionado a evolução da Engenharia de *Software*, criando, para os profissionais do domínio, a necessidade de constante atualização.

Segundo OH e HOEK [5], mesmo com planejamento cuidadoso, nem todos os possíveis eventos podem ser antecipados no início de um projeto de *software*. Durante o ciclo de vida, relações de custo-benefício precisam ser consideradas, conflitos

precisam ser resolvidos e decisões difíceis precisam ser tomadas, o que significa que o desenvolvimento de *software* envolve vários passos intermediários com contínua escolha entre múltiplas alternativas viáveis.

Todos esses aspectos fazem do processo de desenvolvimento de *software* um processo intenso em conhecimento, no qual pessoas altamente qualificadas precisam cooperar entre si para atingir uma meta de negócio comum [6].

Dentro deste contexto, diferentes abordagens têm sido propostas para reduzir custos, encurtar cronogramas e melhorar a qualidade de processos e produtos. No entanto, na Engenharia de *Software* tradicional, pouca atenção tem sido dada à produção, armazenamento, compartilhamento e uso de conhecimentos relevantes para o desenvolvimento de *software*. A consequência é perda de conhecimento, dificuldade de localização e acesso a conhecimentos existentes na organização, além da possível dificuldade de compartilhamento e atualização desses conhecimentos, a depender da mídia e local de armazenamento. Sempre que o conhecimento requerido não está disponível, o desenvolvedor de *software* tem que partir do zero para encontrar a solução do problema, experimentando e construindo novamente o conhecimento, o que tem sérias implicações em termos da qualidade da solução e do tempo e do custo necessários para obtê-la. Recursos anteriormente gastos para obter o conhecimento são novamente requeridos e, pior, erros podem ser repetidos.

STATZ [7] destaca, ainda, que, no ágil mercado contemporâneo, as equipes de projeto necessitam utilizar o tempo para fazer as coisas tão certas quanto possível, pois não há tempo para fazê-las novamente. As equipes necessitam aprender com as experiências de colegas da própria organização e da indústria. Segundo a autora, é difícil para uma equipe de projeto melhorar seu desempenho se ela não refletir sobre o seu trabalho e sobre como melhorá-lo. Neste contexto, algumas equipes refletem sobre seus projetos e os resultados alcançados; algumas equipes compartilham esta informação com outros na organização; mas poucas equipes tentam rever, no início de uma nova fase do projeto ou de um novo projeto, o que outros têm aprendido; e várias equipes não fazem nada disso. Estas últimas são as que enfrentam os mesmos problemas repetidamente, atribuindo-os à falta de sorte, quando, na verdade, a causa básica é a deficiência de aprendizado no projeto e na organização.

1.2 Gerência do Conhecimento

Gerência do Conhecimento aplicada à Engenharia de *Software* busca melhorar o desempenho dos desenvolvedores de *software* ao reconhecer a importância do acesso a dados, informações e conhecimentos quando estes são necessários para completar uma tarefa ou tomar uma decisão. Isto não significa ignorar o valor ou a

necessidade das medidas mais tradicionalmente consideradas, como a introdução de novas tecnologias ou a melhoria de processos, nem tem como objetivo substituí-las [8]. De fato, a Gerência do Conhecimento, ao explicitar e sistematicamente administrar o conhecimento existente na organização, é um instrumento para melhoria dos processos de *software*, uma vez que qualquer melhoria depende do aprendizado em experiências passadas. Quanto à introdução de novas tecnologias, BIRK e KRÖSCHEL [2] argumentam que a falta de informação sobre quando uma tecnologia pode ser aplicada e quando não aplicá-la é a principal razão para muitas falhas de projeto causadas por tecnologia. Além disso, a Gerência do Conhecimento pode ser vista como uma estratégia de prevenção e mitigação de riscos ao, explicitamente, considerar riscos muito freqüentemente ignorados, como perda de conhecimento, falta de conhecimento, re-trabalho, repetição de erros e indisponibilidade de indivíduos que possuem conhecimento-chave para o projeto [8].

Na verdade, conhecimento sempre foi administrado nas organizações em algum nível e de alguma maneira [10,11,12,13]. No entanto, a Gerência do Conhecimento explícita e efetiva tem sido apresentada como um fator chave para o sucesso das organizações nos ambientes de negócio atuais, não sendo diferente quando o negócio é o desenvolvimento de *software* [13]. Segundo KUCZA *et al.* [12], muitas organizações de desenvolvimento de *software* têm reconhecido que, para terem sucesso no futuro, precisam administrar e utilizar conhecimento de forma mais efetiva, produtiva e inovadora, envolvendo desde indivíduos e equipes de projeto à organização como um todo. Assim, as metas gerais da Gerência do Conhecimento podem ser formuladas como: tornar os indivíduos de uma organização conscientes dos processos de conhecimento, melhorar esses processos onde problemas tenham sido identificados, e apoiá-los com os meios técnicos sempre que possível e razoável.

PRAHALAD e HAMEL [14] argumentam que a Gerência do Conhecimento apóia a tomada de decisão eficaz e a criação de soluções inovadoras e criativas, combate a perda de conhecimento por aposentadoria ou migração de especialistas, combate a repetição de erros através da exploração das experiências adquiridas em projetos anteriores, permite que não especialistas obtenham conselho especializado quando necessário, reduz esforço duplicado por eficientemente possibilitar a construção sobre trabalho prévio, além de apoiar a consolidação de conhecimentos espalhados ao longo da organização e a diminuição da curva de aprendizado de novas tecnologias, tornando a organização capaz de adaptar-se mais rapidamente a novas oportunidades.

Por fim, JURISTO e MORENO [4] relembram que, na verdade, o desenvolvimento de *software* ainda tem que atingir o nível de Engenharia e que, para

isto, é necessário identificar relações de causa e efeito que permitam explorar a eficácia das tecnologias de forma quantitativa e repetitiva, citando PFLEEGER [15]: “se observarmos por tempo suficiente e com atenção suficiente, nós iremos achar regras racionais que nos mostrem as melhores formas de construir o melhor *software*”. Neste sentido, a Gerência do Conhecimento, que busca identificar o que funciona no contexto da organização, é capaz de fornecer valioso material para a derivação de conclusões no nível de indústria e do domínio de conhecimento.

1.3 Ambientes de Desenvolvimento de *Software* Orientados a Organização

A questão que surge, então, é como, de fato, aplicar Gerência do Conhecimento em organizações que desenvolvem e mantêm *software*.

Desenvolvedores de *software* produzem e precisam de conhecimento quando estão executando as tarefas pelas quais são responsáveis e vários autores [6,16,17,18,19] ressaltam que a Gerência do Conhecimento deve estar embutida nos processos de trabalho.

Ambientes de Desenvolvimento de *Software* (ADS) são sistemas computacionais que visam apoiar, tanto quanto possível, os desenvolvedores de *software* na execução de suas tarefas durante projetos de desenvolvimento ou manutenção de *software*. Desta forma, constituem-se no ambiente ideal para aplicação das idéias de Gerência do Conhecimento, de forma que o conhecimento adquirido ao longo de vários projetos possa estar disponível para o desenvolvedor de *software* dentro do seu contexto de trabalho e que este possa contribuir diretamente para a geração de novo conhecimento a ser utilizado em outros projetos.

Conhecimento sobre o domínio da aplicação é uma parcela do conhecimento necessário para os desenvolvedores de *software*, cuja integração em ADS foi tratada por OLIVEIRA [20], que propôs os Ambientes de Desenvolvimento de *Software* Orientados a Domínio (ADSOD). A proposta agora é possibilitar a gerência completa do conhecimento que pode ser útil para os desenvolvedores de *software* ao longo dos processos de desenvolvimento e manutenção de *software*, o que também envolve diretrizes e melhores práticas organizacionais, conhecimento sobre técnicas e métodos de desenvolvimento de *software*, relatos de experiências anteriores com o uso destas técnicas e métodos bem como com os processos de *software*, entre outros.

Desta forma, definimos os Ambientes de Desenvolvimento de *Software* Orientados a Organização (ADSOrg) [21], cuja finalidade é fornecer qualquer conhecimento acumulado pela organização que possa ser relevante para os seus

projetos de *software*, além de apoiar, a partir do aprendizado nos projetos específicos, o aprendizado organizacional em Engenharia de *Software*.

ADSOrg pretendem apoiar o desenvolvimento e a manutenção de *software* tanto em organizações em que estas atividades são atividades de negócio, quanto em organizações que possuem outro tipo de negócio e nas quais o desenvolvimento e a manutenção de *software* são atividades de apoio ao negócio. Gerência do Conhecimento aplicada à Engenharia de *Software* é fundamental para ambos os tipos de organizações, pois, no primeiro tipo, o conhecimento acumulado ao longo do tempo no desenvolvimento e na manutenção de *software* serve de base para a criatividade e a inovação em termos de produtos e/ou serviços e, no segundo tipo, as aplicações de *software* estão no caminho crítico de quase todas as atividades organizacionais.

A Estação TABA foi definida e construída na COPPE/UFRJ para auxiliar na definição, implementação e execução de Ambientes de Desenvolvimento de *Software*, tendo sido estendida ao longo dos últimos dez anos [20,22,23]. Para a construção de ADSOrg, a infra-estrutura fornecida pela Estação TABA foi novamente estendida e utilizada.

Com o intuito de avaliar a viabilidade da solução proposta para a construção de ADSOrg, a Estação TABA foi configurada para a Fundação Bahiana de Cardiologia (FBC), organização que desenvolve, através do seu Núcleo de Pesquisa e Desenvolvimento de Software Médico, diferentes tipos de *software*, tais como: sistemas baseados em conhecimento, *software* educacional para medicina, base de dados de pesquisa, sistemas de informação hospitalar, prontuário médico eletrônico e aplicações de telemedicina.

1.4 Histórico de Trabalho

Esta seção apresenta, brevemente, o histórico do trabalho realizado com o intuito de definir e construir Ambientes de Desenvolvimento de Software Orientados a Organização (ADSOrg).

➤ Março de 1999 a Fevereiro de 2000

Este período teve como objetivo a preparação para o Exame de Qualificação através da elaboração da proposta de tese, tendo sido decisiva a participação na 11th *International Conference on Software Engineering & Knowledge Engineering* (SEKE1999) realizada em Kaiserslautern, na Alemanha. A partir das questões discutidas na conferência, das disciplinas cursadas e de revisão bibliográfica, decidiu-se investigar os conceitos de Gerência do Conhecimento e Fábrica de Experiências no contexto dos Ambientes de Desenvolvimento de *Software*. O

resultado foi a definição de requisitos para os ADSOrg e da infra-estrutura necessária para atender a estes requisitos.

➤ **Março de 2000 a Fevereiro de 2001**

Este período, iniciado com o Exame de Qualificação, teve como objetivo principal a discussão da proposta de tese com a professora Shari Pfleeger durante o *Workshop* de Teses e Dissertações em Engenharia de *Software* ocorrido em João Pessoa. As atividades do período consistiram de uma visita à Universidade de Maryland, da participação na *12th International Conference on Software Engineering & Knowledge Engineering (SEKE2000)*, realizada em Chicago, e do início do planejamento de uma pesquisa para identificação dos tipos de conhecimento considerados mais importantes pelos desenvolvedores de *software*, o que incluiu a definição de uma primeira versão do questionário a ser utilizado. As publicações do período foram:

- VILLELA, K., TRAVASSOS, G. H., ROCHA, A. R., “Ambientes de Desenvolvimento de Software Orientados a Organização”, Relatório Técnico da COPPE/UFRJ - ES530/00, 2000;
- VILLELA, K., TRAVASSOS, G. H., ROCHA, A. R., “*Toward Enterprise-Oriented Software Development Environments*”, XIV SBES - Workshop de Teses e Dissertações em Engenharia de Software, João Pessoa, Brasil, 2000.

➤ **Março de 2001 a Fevereiro de 2002**

Neste período, o trabalho de construção da infra-estrutura de conhecimento para os ADSOrg foi iniciado, tendo sido definidos: 1) uma versão preliminar da ontologia de organização¹, que foi amplamente discutida com os principais membros do projeto “Ambientes de Desenvolvimento de *Software* Orientados a Organização”; 2) os requisitos de uma ferramenta capaz de representar a estrutura organizacional e a distribuição de competências ao longo desta estrutura, que deu origem a uma tese de mestrado da COPPE/UFRJ [25]; e 3) os requisitos de uma ferramenta para representação dos processos organizacionais e da distribuição de competências ao longo desses processos, que se tornou objeto de um projeto de final de curso da UFBA [26]. Três outras teses de mestrado foram iniciadas tendo como objetivo apoiar atividades específicas do desenvolvimento de *software* com conhecimento organizacional relacionado às mesmas [27,28,29]. Além das definições mencionadas, o trabalho realizado no período envolveu: 1) o refinamento do questionário voltado

¹ Uma organização é um grupo de indivíduos trabalhando em conjunto para a realização de uma missão, propósito ou objetivo comum [24]. Empresas privadas, empresas e órgãos públicos, organizações não governamentais são alguns dos exemplos.

para a identificação da importância atribuída aos diversos conhecimentos a partir dos comentários de Shari Pfleeger e da avaliação do mesmo por um grupo de desenvolvedores, 2) a orientação do projeto de final de curso da UFBA, que resultou em uma notação para modelagem de processos intensos em conhecimento e em um protótipo da ferramenta, e 3) o apoio às teses de mestrado mencionadas. A notação foi apresentada na COPPE/UFRJ, discutida e revisada, passando a ser utilizada para modelagem dos processos apoiados por diversas teses da linha de Engenharia de Software [27,28,29,30,31,32,33,34,35,36]. No período, foram publicados:

- VILLELA, K., TRAVASSOS, G. H., ROCHA, A. R., “Ambientes de Desenvolvimento de Software Orientados a Organização”, *4º Workshop Ibero-Americano de Ingeniería de Requisitos y Ambientes de Software*, San Jose, Costa Rica, 2001;
- VILLELA, K., SANTOS, G., GALOTTA, C., *et al.*, “Estendendo a Estação TABA para a criação de Ambientes de Desenvolvimento de Software Orientados a Organização”, *XV SBES – Caderno de Ferramentas*, Rio de Janeiro, Brasil, 2001;
- VILLELA, K., OLIVEIRA, K., SANTOS, G., *et al.*, “CORDIS-FBC: Um Ambiente de Desenvolvimento de Software para Cardiologia”, *I Workshop de Informática Médica*, Rio de Janeiro, Brasil, 2001;
- VILLELA, K., ZLOT, F., SANTOS, G., *et al.*, “Knowledge Management in Software Development Environments”, *14th International Conference - Software & Systems Engineering and their Applications*, Paris, França, 2001.

➤ **Março de 2002 a Fevereiro de 2003**

Este período teve como objetivo: 1) estender a Estação TABA para torná-la capaz de gerar ADSOrg; 2) definir um processo padrão de desenvolvimento para a FBC e os processos especializados necessários para a configuração de um ambiente para a organização; 3) avaliar, através da descrição de uma organização (a FBC), a adequação da ontologia de organização e da notação para modelagem de processos; e 4) aplicar, em Salvador, a pesquisa para identificação dos conhecimentos considerados mais importantes nos ambientes organizacionais para desenvolvimento e manutenção de *software*. A extensão da Estação TABA envolveu: 1) o estudo do modelo de dados, da infra-estrutura e das ferramentas disponíveis; 2) a identificação e correção de problemas e melhorias, de forma a obter uma configuração base para a extensão; e 3) a alteração da infra-estrutura, de forma a permitir a configuração de um ambiente para a FBC, que foi demonstrado em setembro, na reunião do Programa Brasileiro de Qualidade e Produtividade. Os ajustes necessários à ontologia de organização foram feitos a partir do uso da mesma para descrição da FBC e da interação com um especialista na construção de ontologias. Para facilitar a execução

da pesquisa, foi desenvolvido um sistema para aplicação de questionários na Web. Neste período, o apoio às teses de mestrado se intensificou e mais 3 teses foram iniciadas no contexto do projeto [30,31,37], tendo sido realizadas várias reuniões para discussão e integração de modelos e idéias. O período foi encerrado com a especificação da versão operacional da ferramenta de modelagem de processos intensos em conhecimento. As publicações do período foram:

- VILLELA, K., TRAVASSOS, G. H., ROCHA, A. R., *et al.*, "The Definition of Software Processes and their Automated Support, taking Domain Knowledge and Organizational Culture into account", *24th ICSE - Workshop on Software Quality*, Orlando, EUA, 2002;
- VILLELA, K., OLIVEIRA, K., SANTOS, G., *et al.*, "Melhoria de Processos de Software e Evolução de Ambientes de Desenvolvimento de Software com base no Conhecimento do Domínio e na Cultura Organizacional", *I Simpósio Brasileiro de Qualidade de Software*, Gramado, Brasil, 2002;
- VILLELA, K., SANTOS, G., TRAVASSOS, G. H., *et al.*, "Gestão de Conhecimento em Ambientes de Desenvolvimento de Software", *2^a Jornadas Ibero-Americanas de Ingeniería del Software e Ingeniería del Conocimiento*, Salvador, Brasil, 2002;
- SANTOS, G., VILLELA, K., TRAVASSOS, G. H., *et al.* "Representing and Visualizing the Distribution of Knowledge, Skills and Experiences throughout the Organizational Structure", *15th International Conference - Software & Systems Engineering and their Applications*, Paris, França, 2002.

➤ **Março de 2003 a Fevereiro de 2004**

O último período foi dedicado: 1) à implementação da ferramenta de modelagem de processos, 2) à análise dos dados coletados na pesquisa, 3) à definição de um processo para orientar a configuração da Estação TABA para uso nas organizações, além da implementação do apoio a este processo, e 4) à redação do texto da tese. O apoio ao processo de configuração resultou em ajustes e redefinição de algumas ferramentas da Estação TABA. Neste período, foram publicados os artigos:

- VILLELA, K., OLIVEIRA, K., SANTOS, G., *et al.*, "Cordis-FBC: an Enterprise-Oriented Software Development Environment", *Workshop on Learning Software Organizations*, Luzern, Suíça, 2003.
- SANTOS, G., VILLELA, K., ROCHA, A. R., "SAPIENS: uma ferramenta para descrição e recuperação de competências em uma organização", *I Workshop de Tecnologia da Informação e Gerência do Conhecimento*, Fortaleza, Brasil, 2003.

1.5 Organização da Tese

Esta tese está organizada em oito capítulos, incluindo este primeiro capítulo introdutório.

No *Capítulo 2 - Gerência do Conhecimento em Engenharia de Software*, são apresentados conceitos de Gerência do Conhecimento e abordagens, sistemas, recomendações e questões relacionados a sua aplicação na Engenharia de Software.

No *Capítulo 3 - Gerência do Conhecimento Orientada a Processos*, a abordagem de embutir Gerência do Conhecimento nos processos organizacionais é discutida no contexto da Engenharia de *Software*, sendo apresentado alguns sistemas e relatos de experiência.

O *Capítulo 4 - Ambientes de Desenvolvimento de Software* apresenta os conceitos de Ambiente de Desenvolvimento de Software Centrado em Processo e de Ambiente de Desenvolvimento de Software Orientado a Domínio, mencionando alguns projetos conduzidos na área e descrevendo, em mais detalhes, o Projeto TABA.

No *Capítulo 5 - Ambientes de Desenvolvimento de Software Orientados a Organização*, a abordagem proposta para introdução de Gerência do Conhecimento em Ambientes de Desenvolvimento de Software é descrita, sendo fornecidos o conceito e os requisitos de Ambientes de Desenvolvimento de Software Orientados a Organização, além de um modelo que define a infra-estrutura necessária em tais ambientes e propostas para a representação dos conhecimentos considerados básicos na abordagem.

O *Capítulo 6 - Pesquisa sobre a Importância dos Tipos de Conhecimento* apresenta os resultados da pesquisa que procurou identificar os tipos de conhecimento considerados mais importantes por desenvolvedores de *software*, de forma que estes possam ser priorizados nos esforços para construção de Ambientes de Desenvolvimento de Software Orientados a Organização.

O *Capítulo 7 - Construção de Ambientes Orientados a Organização na Estação TABA* descreve como a Estação TABA foi estendida para contemplar a construção de Ambientes de Desenvolvimento de Software Orientados a Organização. A execução do processo e das atividades propostas é ilustrada no contexto de uma organização, evidenciando o apoio fornecido pelas ferramentas desenvolvidas.

Por fim, no *Capítulo 8 - Conclusão e Perspectivas Futuras*, as conclusões, perspectivas para continuidade deste trabalho e futuras pesquisas são apresentadas.

Capítulo 2 - Gerência do Conhecimento em Engenharia de Software

2.1 Introdução

O termo Gerência do Conhecimento surgiu no meio dos anos 80, como consequência do reconhecimento da importância do conhecimento para a sobrevivência e o sucesso das organizações. Apesar da idéia de capturar conhecimento e utilizá-lo para auxiliar na execução de tarefas não ser nova, a Gerência do Conhecimento tem foco no indivíduo como especialista e fonte de conhecimento que ele sistematicamente compartilha com a organização [8]. Considerada uma disciplina emergente [8,19,38], muito tem sido discutido sobre Gerência do Conhecimento, mas ainda não existe uma base de definições amplamente aceitas.

Este capítulo apresenta os conceitos que foram considerados mais adequados, as idéias surgidas e as questões que permanecem abertas ao longo dos estudos já realizados, focalizando, sempre que pertinente, no contexto da Engenharia de Software.

2.2 Conceito de Conhecimento

Segundo MARKKULA [39], é essencial entender a diferença entre dado, informação e conhecimento, pois o desempenho de uma organização depende freqüentemente dela saber reconhecer o que necessita, o que já possui e o que é possível fazer com cada um deles. Desta forma, DAVENPORT e PRUSAK [40] definem:

- Dados são um conjunto de fatos discretos e objetivos sobre eventos. O dado não diz nada sobre sua importância ou relevância, mas é o material básico para a criação da informação;
- Informação é uma mensagem com emissor e receptor, que tem a finalidade de mudar a maneira como o receptor percebe algo, ter impacto no seu julgamento ou comportamento. Dado transforma-se em informação quando é adicionado significado;
- Conhecimento é informação combinada com experiência, contexto, interpretação e reflexão. É a forma de informação que está pronta para ser aplicada em decisões e ações.

Além dessas definições, algumas considerações que esclarecem a natureza do conhecimento são:

- Conhecimento inclui abstrações e generalizações de grande volume de informações, sendo tipicamente menos preciso e mais difícil de ser verificado objetivamente [41];
- Conhecimento evolui em estágios: conjecturas e especulações, declarações ainda questionadas, declarações não mais questionadas e, por fim, fatos estabelecidos e aceitos [4].

2.3 Tipos de Conhecimento

Quanto à natureza, o conhecimento pode ser explícito ou tácito [42]. Conhecimento explícito encontra-se expresso através de palavras, símbolos e/ou números, podendo ser facilmente transmitido e compartilhado na forma de procedimentos codificados, fórmulas, princípios, entre outros. Conhecimento tácito é o conhecimento entendido e que pode ser aplicado, mas que ainda não se encontra expresso em palavras, símbolos e/ou números. Muitas vezes, isto ocorre em decorrência do conhecimento ser altamente pessoal e difícil de formalizar, o que o torna também mais difícil de ser compartilhado com outras pessoas.

Segundo RUS e LINDVALL [8], conhecimento também pode ser classificado de acordo com o escopo, tendo-se conhecimento individual, de grupo, organizacional, multi-organizacional ou de indústria. Conhecimento organizacional é conhecimento explícito ou tácito que a organização possui sobre processos, procedimentos, mercados, produtos e tecnologias e que são capazes de gerar lucro e torná-la competitiva [43]. Conhecimento multi-organizacional é freqüentemente requerido em processos de desenvolvimento com inovação a curto prazo quando as organizações estão incertas de como a nova tecnologia irá afetar a indústria [44].

2.4 Processo de Conversão e Evolução do Conhecimento

Conhecimento é criado e expandido através da interação contínua e social entre conhecimento tácito e conhecimento explícito. Através desta interação, conhecimentos tácito e explícito expandem-se em termos de qualidade e quantidade [39]. O processo de conversão e evolução do conhecimento é uma espiral formada por socialização, externalização, combinação e internalização. Durante a socialização, um indivíduo compartilha conhecimento tácito diretamente com um outro indivíduo. Externalização acontece quando o indivíduo é capaz de converter conhecimento tácito em explícito, permitindo que o conhecimento seja compartilhado com o grupo. Já a combinação de

conhecimento explícito dá origem a novo conhecimento explícito. Neste sentido, conhecimentos de vários grupos podem ser combinados de forma a dar origem a conhecimento organizacional. Por fim, na medida em que novo conhecimento explícito é compartilhado, indivíduos utilizam este conhecimento para ampliar e rever seu próprio conhecimento tácito, o que constitui a internalização [45].

MARKKULA [39] observa, ainda, que novo conhecimento está sempre relacionado a um conhecimento já existente, de modo que o aprendizado é cumulativo e o desempenho do aprendizado é melhor quanto mais o objeto de aprendizado estiver relacionado a algo já conhecido.

2.5 Conceito de Gerência do Conhecimento

Conhecimento tem sido considerado como o patrimônio mais importante de uma empresa, possuindo influência decisiva em sua competitividade [8,12,46,47,48,49]. No entanto, para a vantagem competitiva fornecida pelo conhecimento ser sustentável, o conhecimento não pode permanecer no nível individual [50]. Conhecimento individual é perdido quando o indivíduo sai da empresa e pode ser difícil de ser localizado e compartilhado. Conhecimento organizacional também pode representar um problema quando não disponível de forma adequada, de modo que possa ser facilmente acessado, compartilhado e atualizado. Desta forma, as organizações têm adotado a Gerência do Conhecimento como forma de preservar e expandir conhecimento individual, sempre que possível transformando-o em conhecimento organizacional prontamente disponível para os demais indivíduos da organização.

De maneira ampla e como uma síntese de várias definições [39,49,51,52,53,54,55,56,57,58,59], tem-se que Gerência do Conhecimento é a administração, de forma sistemática e ativa, dos recursos de conhecimento de uma organização, utilizando tecnologia apropriada e visando fornecer benefícios estratégicos à organização, o que envolve:

- comunicação efetiva e fluxo de conhecimento entre os indivíduos da organização,
- captura, organização e armazenamento de conhecimento relevante a partir de fontes internas e/ou externas disponíveis para a organização,
- disponibilização e distribuição do conhecimento capturado de forma adequada às necessidades dos usuários,
- desenvolvimento de novos conhecimentos,
- atualização de conhecimento defasado.

Vários autores [19,38,47,53,60,61,62] destacam que a maneira como o conhecimento é administrado depende das metas estratégicas da organização.

Além disso, tanto fatores técnicos quanto fatores não-técnicos influenciam a Gerência do Conhecimento em uma organização. JONES [52] indica os fatores humanos, os fatores sociais, a cultura e os processos organizacionais como fatores não técnicos da Gerência do Conhecimento, que são considerados por SKYRME [63] como fatores determinantes, em última instância, do sucesso de sua implementação em uma organização. Vários outros autores [10,19,39,50,55,57,58,64,65] mencionam fatores que capacitam a Gerência do Conhecimento em uma organização, os quais são discutidos ao longo do capítulo.

2.6 *Frameworks* de Gerência do Conhecimento

Diante dos fatores técnicos e não técnicos envolvidos e do reconhecimento de que não existe uma solução de Gerência do Conhecimento que seja adequada a todas as organizações [47,66], verificou-se a necessidade de definição de um *framework* que contenha os elementos essenciais de um programa de Gerência do Conhecimento de sucesso [47,67]. O objetivo é auxiliar as organizações na definição de suas próprias soluções de Gerência do Conhecimento.

TRITTMANN [67] propõe um *framework* para Gerência do Conhecimento em Engenharia de Software com o objetivo de discutir as diferentes abordagens existentes e seus benefícios, bem como permitir a comparação sistemática de soluções de Gerência do Conhecimento adotadas por diferentes empresas. O *framework* proposto tem quatro dimensões: infra-estrutura técnica, estrutura organizacional, coordenação e motivação.

A dimensão de infra-estrutura técnica é representada pelos sistemas baseados em Tecnologia da Informação. Nesta dimensão, sistemas que apóiam a gerência do conhecimento na sua forma codificada podem ser diferenciados dos sistemas que estão principalmente voltados para a forma personificada do conhecimento¹.

A dimensão da estrutura organizacional trata da atribuição de tarefas e autoridade às unidades organizacionais. De acordo com esta dimensão, tem-se a estrutura centralizada e a estrutura descentralizada. A estrutura centralizada é caracterizada pela atribuição funcional de tarefas a pelo menos um departamento responsável pela Gerência do Conhecimento, que supervisiona em algum grau a organização de desenvolvimento operacional. Na estrutura descentralizada, posições

¹ No primeiro caso, conhecimento é separado do seu repositório original, a mente dos indivíduos, e é transferido para uma forma escrita. No segundo caso, conhecimento é compartilhado pessoalmente, seja face a face ou através de mídias que permitem alto grau de interação entre as pessoas [67].

são criadas para que pessoas distribuídas na organização atuem com assessores para os desenvolvedores de software, fornecendo ajuda sobre tópicos específicos.

Coordenação, a terceira dimensão do *framework*, refere-se à necessidade de coordenar o fornecimento e a demanda de conhecimento dentro da organização que desenvolve software. Neste contexto, a coordenação por padronização é baseada na definição antecipada dos processos de conhecimento e na transferência de conhecimento através de uma posição hierárquica superior, enquanto a coordenação por ajuste mútuo baseia-se na criação de situações em que a transferência de conhecimento acontece espontaneamente e os desenvolvedores de software transferem conhecimento diretamente de um para outro.

Motivação, a última dimensão, está relacionada ao desejo de compartilhar conhecimento com outras pessoas e utilizar conhecimento de outras pessoas na execução de suas próprias tarefas. A motivação intrínseca, que ocorre quando uma tarefa é executada buscando-se satisfação imediata das próprias necessidades, pode ser diferenciada da motivação extrínseca, segundo a qual alguém executa uma atividade por ter sido incentivado a fazê-lo através de, por exemplo, mecanismos de reconhecimento ou compensação. TRITTMANN [67] enfatiza que há estágios intermediários entre as opções propostas para cada dimensão.

A questão, então, é como usar este *framework* para projetar uma solução de Gerência do Conhecimento adequada à organização. TRITTMANN [67] sugere que seja considerado o efeito desejado da solução: 1) inovação, que significa o desenvolvimento de novo conhecimento, necessário para resolver problemas ou dominar tarefas que são novas para a organização; ou 2) otimização, que significa a utilização do conhecimento existente na organização para executar tarefas similares ou produzir melhores soluções. O efeito de otimização está relacionado com sistemas que apóiam a transferência de conhecimento codificado, unidades centralizadas para gerência do conhecimento, coordenação por padronização e motivação extrínseca para o compartilhamento de conhecimento. Já o efeito de inovação está relacionado a sistemas que apóiam a transferência de conhecimento personalizado, estrutura descentralizada, coordenação através de ajuste mútuo e motivação intrínseca. TRITTMANN [67] sugere ainda que o efeito desejado (inovação ou otimização) pode ser diferente para cada área de conhecimento que é comumente considerada importante para o desenvolvimento de software (conhecimento sobre o domínio de aplicação, conhecimento sobre a tarefa ou metodológico, e conhecimento técnico especializado) e que, mesmo buscando o efeito de otimização, uma organização pode adotar transferência de conhecimento personalizado devido à dificuldade de codificar conhecimento complexo.

RUBENSTEIN-MONTANO *et al.* [38] analisam os *frameworks* de Gerência do Conhecimento existentes até então e fornecem as seguintes diretrizes para o desenvolvimento de um *framework* genérico condizente com o pensamento sistêmico²:

- o *framework* deve incluir tanto elementos prescritivos quanto descritivos, o que significa considerar tanto as atividades necessárias para a Gerência do Conhecimento quanto as partes adicionais do sistema que têm impacto nessas atividades, tais como objetivos de negócio, cultura/pessoas e aprendizado;
- o *framework* deve incluir planejamento para definir o problema/objetivo de uma organização antes de considerar as atividades de Gerência do Conhecimento, o que significa identificar o propósito da Gerência do Conhecimento de acordo com as metas e estratégias organizacionais;
- o *framework* deve integrar estratégias voltadas para a gerência do conhecimento codificado e personalizado em uma abordagem sistêmica que considere tanto o ciclo de conhecimento quanto o ambiente cultural. Tecnologias, métodos e ferramentas usados para Gerência do Conhecimento deveriam ser estabelecidos de acordo com a cultura organizacional;
- o *framework* deve considerar a Gerência do Conhecimento como um processo evolutivo e de melhoria contínua que inclui múltiplos ciclos de *feedback* e aprendizado no qual novo conhecimento é sintetizado através da combinação do conhecimento existente de maneira inovadora (*double-loop learning*).

Segundo os autores, estas recomendações são complementares às comumente feitas, que se referem à prescrição de atividades de Gerência do Conhecimento, tratamento apropriado de conhecimento explícito e tácito, e aprendizado com propósitos corretivos, ou seja, aprendizado que introduz mudanças incrementais (*single-loop learning*). Os autores concluem que as dimensões de tal *framework* poderiam ser: objetivo estratégico da organização, conhecimento (tipos, fluxo e tarefas), aprendizado, tecnologia e subgrupos de pessoas/cultura dentro da organização.

Além dos *frameworks* analisados por RUBENSTEIN-MONTANO *et al.* [38], é possível, ainda, citar os *frameworks* definidos por PROBST *et al.* [68] e SNOEK [69]. A finalidade do *framework* definido por SNOEK [69] é possibilitar a comparação das várias soluções de Gerência do Conhecimento. O *framework* é uma extensão do *framework* definido por PROBST *et al.* [68]³, sendo bastante detalhado ao definir

² Pensamento sistêmico é uma abordagem conceitual para a solução de problemas que examina as relações entre as várias partes do sistema, distinguindo o que é parte do sistema e o que é o ambiente do sistema [38].

³ A versão em alemão da referência já estava disponível em 1998.

atributos com respectivos valores possíveis para cada um dos componentes do *framework* (metas, aquisição, processamento, distribuição, utilização e validação). No entanto, o *framework* é voltado para a comparação da infra-estrutura técnica das soluções de Gerência do Conhecimento. No *framework* proposto por PROBST *et al.* [68], o processo de aquisição é dividido em identificação e coleta de conhecimento.

FELDMANN e ALTHOFF [70] classificam as abordagens baseadas principalmente na colaboração e comunicação como abordagens orientadas a processo, e as abordagens baseadas principalmente na documentação, armazenamento e reutilização de conhecimento organizacional como abordagens orientadas a produto, que é exatamente o que TRITTMANN [67] pretende capturar na dimensão de infra-estrutura técnica do seu *framework* e SNOEK [69] pretende capturar com o atributo “Principal Orientação”.

O’LEARY [71] defende que a freqüência de reutilização do conhecimento seja utilizada para definir as características dos Sistemas de Gerência do Conhecimento e avaliar o quão bem o conhecimento tem sido gerenciado. Sistemas projetados para facilitar a reutilização freqüente de conhecimento devem estar estreitamente integrados com os processos de negócio, o que torna mais fácil prever quais são os recursos de conhecimento necessários. Já sistemas projetados para uso não freqüente de recursos de conhecimento devem ser independentes de um processo específico e facilitar a navegação entre o conhecimento existente, que deve ser mais amplo para que seja possível encontrar conhecimento não especificado previamente. No que se refere à avaliação, outros fatores precisam também ser considerados, como o domínio de conhecimento, que pode tornar a reutilização freqüente de conhecimento impraticável ou mesmo não desejada, e a cultura organizacional, que pode inibir o compartilhamento de conhecimento. Apesar de O’LEARY [71] atribuir esta característica a sistemas, a freqüência de reutilização do conhecimento pode variar no contexto de um mesmo sistema.

2.7 Fábrica de Experiência: o mais popular dos *frameworks*

O conceito de Fábrica de Experiências [72] surgiu como uma abordagem para melhorar a qualidade de processos e de produtos em organizações que desenvolvem e mantêm software através da definição de um ambiente para experimentação controlada, análise de processos de desenvolvimento, aprendizado e reutilização de experiências. A abordagem, que se tornou a mais popular para Gerência do Conhecimento em Engenharia de Software [67], descreve uma estrutura organizacional composta por Organizações de Projeto e Fábrica de Experiências. O objetivo principal de uma Organização do Projeto é fornecer produtos de software de

acordo com os requisitos estabelecidos, enquanto o objetivo da Fábrica de Experiências é, no decorrer do projeto, apoiar a Organização do Projeto com conhecimento e experiência e, ao final do projeto, empacotar a experiência adquirida, armazenando-a em uma base. Neste contexto, experiência é considerada como conhecimento adquirido pela prática [73], o que inclui modelos, especificações, código, materiais de treinamento, melhores práticas em engenharia de software e lições aprendidas [74]. O Paradigma de Melhoria da Qualidade (*Quality Improvement Paradigm*) orienta a melhoria contínua da qualidade e integra a Organização do Projeto com a Fábrica de Experiências [75,76].

Analisando de acordo com as idéias de TRITTMANN [67] e RUBENSTEIN-MONTANO *et al.* [38], a abordagem da Fábrica de Experiências define um *framework* para gerência do conhecimento que prescreve atividades, estrutura organizacional centralizada e coordenação por padronização; contempla conhecimento na forma codificada; trata a questão do aprendizado através da prescrição de dois ciclos de *feedback* que apóiam, respectivamente, aprendizado no nível de projeto e aprendizado organizacional; deixa a motivação para ser definida por cada implementação da abordagem e não considera explicitamente os objetivos de negócio e a cultura organizacional. Sendo assim, algumas evoluções têm sido propostas para a abordagem [75,76,77]. Além disso, implementar uma Fábrica de Software não parece ser a única forma de gerenciar conhecimento relevante para a Engenharia de Software [67].

2.8 Processo de Implementação de Gerência do Conhecimento

Para TIWANA [61], a Gerência do Conhecimento deve ser implantada nas organizações através da definição de uma série de processos para evitar a rigidez de metodológica e o determinismo tecnológico. Desta forma, o autor propõe um processo incremental contínuo, composto de 10 passos organizados em 4 fases, onde cada passo é, na realidade, um processo. A primeira fase, Levantamento da Infra-estrutura, engloba a análise da infra-estrutura de Tecnologia da Informação existente na organização e o alinhamento da Gerência do Conhecimento à estratégia de negócio. Este último é um passo complexo e crítico que implica em adotar uma estratégia de Gerência do Conhecimento com foco no conhecimento personalizado ou codificado de acordo com as características e objetivos da organização. A segunda fase refere-se à Análise, Projeto e Desenvolvimento do Sistema de Gerência do Conhecimento, envolvendo identificar os componentes da infra-estrutura de conhecimento, fazer uma auditoria dos itens de conhecimento existentes na organização, montar a equipe de Gerência do Conhecimento, criar uma arquitetura técnica do Sistema de Gerência do

Conhecimento, desenvolver o sistema de Gerência do Conhecimento. A terceira fase, denominada Disponibilização Incremental Dirigida a Resultados, implica na disponibilização do Sistema de Gerência do Conhecimento em incrementos, antes de estar completamente desenvolvido, e na Gerência de Mudanças, implementação de sistemas de compensação e remoção de barreiras culturais. A abordagem incremental dirigida a resultados assegura que cada componente está produzindo os benefícios esperados antes que o próximo componente seja desenvolvido e disponibilizado. A última fase é a de Avaliação de Desempenho e de Retorno do Investimento, que busca a medição precisa dos benefícios obtidos através de métricas compostas, não tão rígidas quanto a análise de custo-benefício e as estimativas de retorno financeiro. O último passo sinaliza o início de uma nova iteração incremental. TIWANA [61] discute cada um dos passos e fornece um conjunto de perguntas e respostas que auxiliam uma organização a identificar qual é o seu foco para a Gerência do Conhecimento: personalização ou codificação.

TAUTZ [78] propõe um método, chamado DISER (*Design and Implementation of Software Engineering Repositories*), que consiste de: elaborar um modelo do negócio focado nos aspectos de uso e produção de conhecimento e experiências; definir objetivos concretos a serem atingidos pela Fábrica de Experiências, levando em consideração os interesses das pessoas envolvidas e estabelecendo critérios de sucesso apropriados para permitir a medição do progresso em relação ao alcance dos objetivos; identificar e selecionar áreas de assuntos relevantes que contribuem para o alcance dos objetivos; descrever a aquisição e uso de experiências através de cenários, o que permite a identificação mais detalhada da necessidade de informação; desenvolver um modelo conceitual para experiências e, por último, planejar a aquisição de conhecimento em detalhes e descrever um plano de aquisição de conhecimento.

LINDVALL *et al.* [3] apresenta uma metodologia para implementação de um Sistema de Gerência de Experiências em uma organização específica com base na abordagem da Fábrica de Experiências. O primeiro passo da metodologia é caracterizar a organização e definir os atuais processos de negócio e o conhecimento existente, diferenciando conhecimento documentado, não documentado e não disponível. Depois da caracterização da organização, são definidos os papéis dos usuários no sistema com base na cultura organizacional e são desenvolvidos os casos de uso baseados nos processos de negócio e nos papéis desempenhados pelos usuários. O terceiro passo é a definição de uma taxonomia adequada à organização que possa ser utilizada para classificar as experiências que serão incluídas no sistema e, conseqüentemente, facilitar a posterior localização das mesmas. O quarto passo

refere-se a definição da arquitetura do Sistema de Gerência de Experiências, que inclui as aplicações já em uso na organização. As ferramentas de apoio são, então, desenvolvidas, instaladas e integradas. Depois da implementação da arquitetura, um conjunto de procedimentos para a manutenção regular do Sistema de Gerência de Experiências é criado, de forma a assegurar que o sistema funcione adequadamente e a experiência gerenciada esteja sempre atualizada. O próximo passo é o preenchimento da base de experiências com um conjunto inicial de pacotes de experiência e a configuração e instalação do sistema. Um plano de implantação é preparado e executado para treinar, divulgar e motivar as pessoas a utilizarem o sistema. Depois de disponibilizado, o sistema é constantemente melhorado de acordo com o *feedback* recebido.

KUCZA *et al.* [12] propõem começar com um projeto piloto, melhorar os processos de Gerência do Conhecimento até que eles funcionem adequadamente e, então, ampliar a área de cobertura, de forma a evitar uma mudança radical e prover um desenvolvimento contínuo e gradual da Gerência do Conhecimento na organização. Com isto, as pessoas se acostumam com as novas práticas de trabalho, uma nova cultura é estabelecida passo a passo e custos podem ser mantidos em níveis aceitáveis até que o sucesso possa ser verificado e maiores investimentos sejam possíveis. Um ciclo de melhoria dos processos de Gerência do Conhecimento é proposto com esta finalidade.

2.9 Atividades de Gerência do Conhecimento

RUBENSTEIN-MONTANO *et al.* [38], ao analisarem *frameworks* de Gerência do Conhecimento, comparam as atividades⁴ que têm sido propostas para a Gerência do Conhecimento. Os autores ressaltam que há muitas diferenças sutis entre as atividades propostas, resultantes da combinação de atividades em uma única atividade e da ordenação das mesmas.

Vários outros autores, além dos mencionados por RUBENSTEIN-MONTANO *et al.* [38], definem atividades para Gerência do Conhecimento, especialmente para o ciclo de conhecimento, que define como o conhecimento flui e é manipulado na organização. A tabela 2.1 evidencia o relacionamento entre as atividades por eles propostas, visando facilitar a definição de atividades genéricas e em diferentes níveis para compor o *framework* de Gerência do Conhecimento. Desta forma, o conjunto de

⁴ PROBST *et al.* [68] explicita que atividades podem ser consideradas como processos neste contexto, o que explica o fato de alguns autores se referirem a atividades, enquanto outros se referem a processos. Nesta seção, para simplificação, será utilizado o termo atividade.

atividades mais adequado para uma organização poderia ser definido a partir da seleção das atividades prescritas no *framework*.

Tabela 2.1 – Atividades de Gerência do Conhecimento

Atividades/Sub-atividades	Referências	Outros Nomes Utilizados	Referências
Definição de Metas	[68,69]	Coordenação	[12]
Identificação das Necessidades	[12]		
Obtenção de Conhecimento	[10]	Construção de Conhecimento	[79]
Identificação	[46,63,68]		
Aquisição	[8,46,68,69]	Captura	[8,80]
		Coleta	[12,63]
Desenvolvimento	[46,68]	Criação	[8,12,63,50,80]
		Processamento	[69]
Preservação de Conhecimento	[46,68,69]		
Transformação/Organização	[8]	Apresentação	[10]
Armazenamento	[79]		
Atualização	[12]		
Remoção	[10]		
Disseminação/Usos de Conhecimento	[63]		
Disseminação	[46]	Transferência	[10,79]
		Distribuição e Compartilhamento	[50,68]
		Compartilhamento	[12]
		Distribuição	[69]
		Recuperação e Acesso	[80]
		Disponibilização e Acesso	[8]
Uso	[10,46,68,69,80]	Aplicação	[8,50,79]
Avaliação de Conhecimento	[50]		
Avaliação da Infra-estrutura	[68,69]		

Alguns comentários sobre a tabela 2.1 são, no entanto, necessários. STAAB *et al.* [80] mencionam a atividade de Criação e Importação, outros autores [8,12] consideram importação como parte da atividade de Criação, enquanto PROBST [68] e SNOEK [69] propõem uma atividade apenas para a captura de conhecimento externo. Estes dois últimos autores agrupam criação de conhecimento e captura de conhecimento existente na organização em uma mesma atividade, enquanto outros autores [12,50,63] diferem a atividade de gerar conhecimento novo da atividade de simplesmente localizar e obter conhecimento já existente. Já para STAAB *et al.* [80], a atividade de Captura está relacionada à idéia de representação do conhecimento, o que está de acordo com o que SKYRME [63] propõe como Coleta e Codificação. KUCZA *et al.* [12] reúnem Coleta e Armazenamento em uma única atividade. Segundo ALLWEYER [10], a atividade de Obtenção de Conhecimento inclui atualização e a atividade de Apresentação consiste na documentação, edição, estruturação e associação do conhecimento, além da documentação das meta-estruturas. A atividade de Remoção é considerada como parte da atividade de Atualização por KUCZA *et al.* [12] e como parte da atividade de Avaliação de Conhecimento por HENDRIKS e VRIENS [50]. Por fim, as atividades de Definição de Metas, Identificação das Necessidades e Identificação do conhecimento existente são combinadas por

HENDRIKS e VRIENS [50] em identificação e determinação da situação do conhecimento existente, avaliação do valor estratégico, comparação com o acervo de conhecimento das organizações competidoras e determinação do conhecimento requerido.

2.10 Infra-estrutura Técnica de Gerência do Conhecimento

Sistemas de Gerência do Conhecimento representam a infra-estrutura técnica de Gerência do Conhecimento, que é composta de estrutura, conteúdo, procedimentos, além de serviços e ferramentas de software. LINDVALL *et al.* [3] esclarecem que o conteúdo pode ser dado, informação, conhecimento e experiência, enquanto a estrutura é a forma utilizada para organizar o conteúdo. O conteúdo e a estrutura compõem a memória organizacional, que é considerada o núcleo da infra-estrutura técnica por ABECKER *et al.* [46], SCHNEIDER e SCHWINN [81] e LINDVALL *et al.* [3]. Procedimentos são instruções de como gerenciar a memória organizacional no dia a dia, incluindo como usar, empacotar, combinar, excluir e atualizar conhecimento. Serviços e ferramentas de software apóiam a execução dos procedimentos e auxiliam o ciclo de conhecimento. Tecnologias da Informação modernas possibilitam Sistemas de Gerência do Conhecimento avançados, mas o que faz um sistema ser útil e aceito é a compreensão do problema a ser resolvido.

Para ABECKER *et al.* [46,82], um Sistema de Gerência do Conhecimento deve apresentar as seguintes características:

- coleta e organização sistemática de informação de várias fontes para evitar a perda e melhorar o acesso a todos os tipos de conhecimento organizacionais;
- minimização inicial da engenharia do conhecimento, o que significa explorar os conhecimentos já disponíveis, fornecer benefícios rapidamente e adaptar-se a novos requisitos;
- integração com o ambiente de trabalho, observando o usuário na execução de suas atividades e automaticamente coletando e armazenando fatos interessantes;
- apresentação ativa de informação relevante, uma vez que os funcionários das organizações freqüentemente estão muito ocupados para procurar conhecimento ou nem mesmo sabem da existência de conhecimento pertinente;
- integração entre diferentes representações de conhecimento, o que significa a possibilidade de uso e de visão conjunta de diferentes representações de conhecimento (formais, informais e embutidas em artefatos) e dos relacionamentos entre estas representações;

- exploração de *feedback* dos usuários para manutenção e evolução da Memória Organizacional, que devem poder apontar deficiências e sugerir melhorias sem interromper, de forma significativa, seu fluxo de trabalho.

2.10.1 Memória Organizacional

DIENG [57] define Memória Organizacional ou Memória Corporativa como uma representação persistente e explícita das informações e conhecimentos vitais para uma organização, de forma a facilitar o acesso, o compartilhamento e a reutilização dos mesmos pelos membros da organização durante a execução de tarefas individuais ou coletivas. Além disso, a autora identifica diferentes papéis a serem desempenhados pelas pessoas neste contexto: fonte de conhecimento humano, engenheiro do conhecimento, observador do conhecimento (que coleta e distribui conhecimento do ambiente externo), desenvolvedor da Memória Organizacional, gerente da Memória Organizacional, especialista (que compõe uma equipe de referência para validação do conhecimento) e usuários.

Sob outro ponto de vista, a Memória Organizacional pode ser considerada como um repositório do conhecimento disponível na organização, cuja finalidade é assegurar que o conhecimento desejado possa ser recuperado no tempo e no lugar certo [53]. O termo repositório, no entanto, não significa um repositório físico único, monolítico e de algum tipo específico, como observado por ACKERMAN e HALVERSON [83], pois nem mesmo as organizações são entidades monolíticas. Neste sentido, O'LEARY [54] explica que os vários repositórios de conhecimento podem diferir quanto ao conteúdo, quanto a serem utilizados pela máquina ou pelo usuário e quanto à complexidade de desenvolvimento, citando fatores que influenciam na complexidade do desenvolvimento.

A Base de Experiências atua como Memória Organizacional na abordagem da Fábrica de Experiências, mesmo considerando que, como inicialmente concebida, ela contenha apenas experiências⁵. SCHNEIDER e HUNNIUS [84] recomendam que se deve ser específico no desenvolvimento de um repositório de conhecimentos/experiências, apoiando um processo da organização de cada vez, o que torna mais fácil garantir que cada indivíduo que acesse o repositório possa considerá-lo útil. Além disso, os autores fornecem cinco atributos de qualidade a serem considerados no desenvolvimento e/ou na avaliação de um repositório de conhecimentos/experiências: orientação ao usuário, usabilidade, conformidade com o processo, mecanismos de

⁵ Desta forma, o termo Memória Organizacional/Base de Experiências será utilizado sempre que a idéia do autor sobre Base de Experiências for aplicável a ambos os conceitos, preservando a origem da idéia e ampliando o seu contexto de aplicação.

feedback e manutenibilidade. Os três primeiros atributos atendem ao desejo do usuário de utilizar facilmente o repositório, ser guiado para os conhecimentos/experiências de que necessita e ter suas tarefas diretamente apoiadas. Os dois últimos atributos se referem à necessidade do repositório encorajar o *feedback* dos usuários e ser capaz de evoluir, pois conhecimento/experiência necessita contínua atualização. SCHNEIDER e HUNNIUS [84] fornecem, ainda, um *checklist* baseado nesses atributos para ser utilizado na avaliação de repositórios.

➤ Conteúdo

SCHNEIDER e SCHWINN [81] e BASILI *et al.* [66] destacam a importância da Memória Organizacional/Base de Experiência ser, desde o início, preenchida com conteúdo útil para seus usuários. As fontes de conteúdo podem ser tanto internas quanto externas. SCHNEIDER e SCHWINN [81] defendem, ainda, que o conhecimento requerido por usuários com diferentes papéis difere quanto ao conteúdo, formato e apresentação.

Uma abordagem para tornar a Memória Organizacional/Base de Experiência útil, mesmo quando quase vazia, é permitir que o usuário, sempre que não encontrar o conhecimento que necessita, entre em contato com o especialista humano que o possui. Esta abordagem, quando apoiada de forma adequada, tem a vantagem de preencher a Memória Organizacional/Base de Experiência com conhecimento que tem demanda para compartilhamento [66].

O conteúdo que tem sido contemplado pelas soluções de Gerência do Conhecimento em Engenharia de Software é listado na tabela 2.2.

Tabela 2.2 – Conteúdo contemplado em Soluções de Gerência do Conhecimento

Conteúdo	Referências
Modelo do processo de negócio	[3,10,66,74,85]
Modelo do processo de software	[6,58,85,86,87]
Modelo da estrutura organizacional com perfil dos indivíduos e informação de contato, além de nomes de especialistas por domínio	[49]
Informações sobre pessoas (competências, interesses de pesquisa, currículo e <i>links</i> para <i>home page</i>)	[3,10,58,74,88]
Modelo do conhecimento relevante para a organização	[10]
Documentos corporativos	[3]
Endereços de contato	[85]
Notícias de interesse da organização	[58]
Boas ou melhores práticas	[65,81]
Lições aprendidas	[3,17,49,65,74,85,87,89]
Diretrizes, roteiros e <i>checklists</i>	[3,58,81]
Diretrizes, roteiros e <i>checklists</i> relacionados a processo	[88]
Perguntas freqüentes com respectivas respostas	[58,66,81]
Artefatos de projetos anteriores ou de atividades prévias	[3,58,87,88,89]
Informações sobre projetos (parceiros, propostas, financiamentos, produtos)	[3]
Artigos sobre tecnologia, domínio, tendências, cultura, experiência de projeto, literatura interna e externa em formato pré-definidos	[88]
Lista e comentário de <i>sites</i> relevantes	[58]
Material introdutório para iniciantes	[58]

O modelo de processo de negócio proposto por ALLWEYER [10] contém a indicação de como o conhecimento é processado ao longo dos processos, ou seja, inclui o conhecimento exigido e produzido por cada atividade, permitindo que conhecimento documentado seja associado a sua referência no modelo.

Boas e melhores práticas são pouco referenciadas pelos autores, mas isto pode ser explicado pelo fato de bases de melhores práticas serem particularmente difíceis de serem desenvolvidas [54], requerendo primeiro que sejam formuladas lições aprendidas e que estas sejam consolidadas através do uso em vários projetos.

De acordo com BASILI *et al.* [66], a qualidade, a relevância e a atualidade do conteúdo são condições necessárias para o uso da infra-estrutura técnica de Gerência do Conhecimento.

➤ **Estrutura**

Segundo JABLONSKI *et al.* [90], duas questões devem ser observadas quanto à estrutura de uma Memória Organizacional: 1) a informação precisa ser preparada de tal maneira que possa ser compartilhada, o que significa que compartilhamento requer uma forma adequada de representação, 2) a informação deve ser organizada de maneira que possa ser achada quando necessário, o que significa que existe a necessidade de uma estrutura para classificar as informações. Ainda segundo os autores, há duas maneiras de construir uma base de conhecimento. Uma maneira é extrair a informação dos respectivos armazenadores de conhecimento (arquivos de documentos, bases de dados, entre outros), convertendo-a para um formato proprietário e classificando-a para posterior uso. A segunda maneira consiste em organizar os armazenadores de conhecimento na base de conhecimento, diminuindo o trabalho de preparação. Em ambos os casos, é necessário estruturar a base de conhecimento de forma a permitir a busca de conhecimento. A estrutura (ou esquema) da base de conhecimento é construída de acordo com a área de aplicação e também é considerada uma informação importante. A aceitação da Gerência do Conhecimento em uma organização depende da capacidade semântica da estrutura do repositório de conhecimento.

RAMESH [91] observa, ainda, a necessidade de associação dos fragmentos de conhecimento relacionados para facilitar a transferência e a reutilização de conhecimento.

Para complementar, é clara a importância de contextualizar o conhecimento na Memória Organizacional, quer seja por meio de atributos ou de associações. O'LEARY [92] propõe atributos descritivos, tais como nome e função de contato ou de quem fez a contribuição, data de contribuição e situação do item de conhecimento. Para DIENG

[57], a Memória Organizacional pode ser modelada de várias perspectivas, entre as quais: o quê, como, quando, quem, onde e por quê. RAMESH [91] propõe que os objetos produzidos no desenvolvimento de sistemas sejam associados às pessoas responsáveis pela sua criação, manutenção e uso.

Segundo DIENG [57], técnicas para construção de Memórias Organizacionais podem ser não computacionais, baseadas em base de dados, baseadas em documento, baseadas em conhecimento ou baseadas em caso.

No que se refere à classificação, taxonomias, ontologias⁶ e modelos de negócio têm desempenhado papel fundamental. Como ontologias fornecem um vocabulário compartilhado e facilitam o entendimento do universo que modelam, vários autores [16,54,57,80,92,93] têm defendido o seu uso na classificação do conhecimento organizacional, pois, além de auxiliarem na classificação e indexação propriamente ditas, facilitam a busca semântica e a distribuição de conhecimento relevante para o usuário, além de facilitar a comunicação entre múltiplos usuários e a associação entre múltiplas bases de conhecimento. Segundo O'LEARY [54], a ontologia adotada deve permitir aos usuários estabelecer os recursos de conhecimento que precisam e desejam. Neste contexto, RABARIJAONA *et al.* [93] destacam dois tipos de ontologias que podem ser úteis para a Gerência do Conhecimento: ontologias de domínio e ontologias de empresa. Já LINDVALL *et al.* [3] destacam a importância da taxonomia para facilitar e agilizar a busca de conhecimento, defendendo que o entendimento da organização e de seus processos é crucial para a definição de uma taxonomia útil. Taxonomias são mais simples que ontologias, sendo mais fáceis de serem definidas, mas, no entanto, capturam menos semântica. Por fim, HOLZ *et al.* [94] e NAGEL [17] defendem que processos são entidades naturais para gerentes e desenvolvedores de software e que conhecimento é derivado de e associado a processo, de forma que organizar a Memória Organizacional segundo os processos da organização é uma abordagem que pode ser muito simples, além de trazer uma série de vantagens. A abordagem é discutida em detalhes no capítulo 3. DIENG [57], por exemplo, propõe que agentes inteligentes e máquinas de busca sejam apoiados por uma ontologia do domínio ou um modelo da organização para melhorar os mecanismos de disseminação de conhecimento.

As Memórias Organizacionais, considerando principalmente as soluções de Gerência do Conhecimento em Engenharia de Software, têm sido organizadas em:

⁶ Uma ontologia é uma representação explícita de uma conceituação ou de alguma parte de uma conceituação, onde o termo conceituação se refere ao conjunto de conceitos (por exemplo: entidades, atributos, processos), suas definições e inter-relações que descrevem uma visão do mundo dentro de um domínio de interesse [95].

- documentos conectados a dimensões de acordo com o conteúdo e etiquetados com atributos [90];
- documentos com classificação facetada⁷ e palavras-chave, estruturados e associados de acordo com o conteúdo [81];
- documentos anotados com meta-dados de acordo com ontologias [57,93];
- casos indexados por indicadores de conteúdo [58];
- pacotes de experiência e uma coleção de definições e taxonomias [9,73];
- pacotes de experiência associados a sub-processos ou atividades de processos [17];
- itens de conhecimento (documentos, ponteiros para documento ou para conhecimento tácito) anotados de acordo com ontologias e associados entre si [16,46,74].

2.10.2 Procedimentos e Serviços/Ferramentas de Software

Segundo BASILI *et al.* [66], procedimentos e serviços/ferramentas de software têm que ser definidos de forma a interromper o mínimo possível o trabalho das pessoas na organização. HENNINGER [96] acrescenta que a ênfase de ser colocada na definição de procedimentos e de serviços/ferramentas que apoiem as atividades das pessoas distribuindo informação chave quando ela é necessária, e que isto envolve o entendimento das práticas e atividades de desenvolvimento de software. O autor recomenda melhor entendimento da interação entre a exploração de conhecimento existente e a criação de novo conhecimento, além de recomendar o uso de ferramentas que permitam improvisação enquanto apoiam decisões baseadas em melhores práticas já estabelecidas.

DIENG [57] acrescenta que a criação e a evolução da Memória Organizacional pode ser distribuída (por exemplo: cada fonte de conhecimento pode atualizar diretamente a Memória Organizacional) ou centralizada (uma pessoa ou um serviço gerencia a Memória Organizacional), sendo que a colaboração e o compartilhamento são mais incentivados na opção distribuída.

A seguir, novamente com foco na Gerência do Conhecimento em Engenharia de Software, são apresentados procedimentos e serviços/ferramentas de software de acordo com as atividades de Gerência do Conhecimento que apoiam, tendo sido escolhido o conjunto de atividades mais adequado a este objetivo.

⁷ A classificação facetada é feita a partir da combinação de classes elementares que possuem valores provenientes de um vocabulário controlado [97].

➤ **Identificação de Conhecimento**

A identificação de conhecimento tem como objetivo fornecer a localização dos conhecimentos relevantes para uma organização, quer eles sejam internos ou externos. Como nem todo conhecimento tácito de uma organização pode ser explicitado e nem todo conhecimento explícito pode ser documentado [8], é essencial a capacidade de identificar aqueles que possuem o conhecimento necessário para resolver um determinado problema [91].

Os serviços/ferramentas de software que apóiam a identificação de conhecimento têm sido voltados para:

- fornecimento de uma espécie de “páginas amarelas” associada à descrição dos processos de negócio, permitindo localizar pessoas de acordo com os papéis que desempenham nos processos e entrar em contato com elas pessoalmente [85];
- análise das informações contidas na Memória Organizacional (tal como quem tem publicado informação sobre quais tópicos) para apontar produtores de conhecimento para os consumidores [98];
- gerência de competências, que tem como propósito encontrar as pessoas da organização com as competências necessárias para novos projetos, encontrar pessoas que saibam algo que alguém precisa saber, identificar necessidade de desenvolvimento de competências e identificar áreas para novos projetos [8,99].

Segundo DINGSOYR e ROYRVIK [99], para a ferramenta de gerência de competência funcionar, a organização precisa selecionar o conjunto de competências em que está interessada, ter um sistema para avaliar os funcionários e tornar a informação disponível para diferentes grupos de usuários. A atualização pode ser consequência natural do término dos projetos. Como e quem deve avaliar os níveis de competência das pessoas são as grandes questões. Algumas ferramentas deixam os especialistas elaborarem e atualizarem seus próprios perfis (um exemplo é a ferramenta descrita em [99]), outras ferramentas automaticamente geram os perfis assumindo que os e-mails e documentos das pessoas refletem suas competências [8].

➤ **Desenvolvimento de Conhecimento**

Desenvolvimento de conhecimento envolve a criação de conhecimento e a captura de conhecimentos já existentes na organização [68].

Os serviços/ferramentas de software que apóiam o desenvolvimento de conhecimento têm sido voltados para:

- apoio à solução de problemas relatados por indivíduos da organização através da discussão dos casos relevantes entre especialistas selecionados por gerentes. O

sistema envia o conhecimento recentemente criado para potenciais usuários, o indivíduo que relatou o problema e os especialistas que o resolveram [49];

- apoio à discussão de idéias entre membros da organização que compõem uma comunidade da prática. Questões que não puderam ser respondidas por outra fonte de informação são submetidas como tópicos de discussão [85];
- submissão de perguntas para especialistas, sendo as perguntas e respostas automaticamente capturadas, analisadas e adicionadas à Memória Organizacional [58,66];
- apoio à execução, de forma cooperativa, de atividades inevitáveis (priorização de riscos, por exemplo) e registro do trabalho cooperativo (espaço de trabalho compartilhado e comunicação) para posterior visualização [100];
- apoio a apresentações de projeto que, através de *brainstorming* estruturado, possibilitem a extração de lições aprendidas [66];
- captura dos conceitos e raciocínio embutidos em um protótipo durante a sua demonstração, fornecendo um *checklist* com as questões a serem consideradas em uma boa demonstração [100];
- captura de diálogos relacionados a problemas técnicos que, depois de analisados, são adicionados à Memória Organizacional [8,66];
- apoio à coleta de conhecimento através de formulários *on-line* [17,58,101].

Os seguintes procedimentos têm sido propostos visando o desenvolvimento de conhecimento:

- uso de Pacotes de Observação, que são formulários de uma página para capturar experiências durante a execução das atividades, contendo questões que as pessoas podem e querem responder [100];
- técnica LIDs (*Light-Weight Approach to Experience Elicitation and Reuse*), utilizada para capturar experiência de grupos pequenos, contendo um processo e uma tabela de conteúdo para o documento a ser derivado. A tabela de conteúdo atua como um *checklist* e guia a reflexão do grupo [100];
- captura de conhecimento a partir da identificação das necessidades imediatas dos projetos. Para isto, um projeto para captura de conhecimento é estabelecido; as necessidades dos projetos clientes são identificadas em termos do conhecimento requerido e do tipo (relacionado a produto ou processo), formato e situação de reutilização; e entrevistas estruturadas são realizadas [102];
- Revisão *Postmortem* Simplificada para coleta de experiência de projetos. O método é aplicado em um encontro e consiste de uma introdução, duas sessões de *brainstorm* dedicadas, respectivamente, ao que deu certo e aos problemas que

foram percebidos no projeto, e uma análise das causas dos principais pontos. Um relatório é escrito por pesquisadores e apresentado aos participantes no dia seguinte para obter *feedback* e correções [103];

- Relatório de Experiências redigido pelo gerente para coleta de experiência de projeto. O relatório segue um formato pré-determinado para permitir comparação com relatórios de diferentes projetos, contendo visão geral do projeto, além de descrição de problemas e proposta de melhoria para cada fase do projeto [103].

DINGSOYR *et al.* [103] comparam os métodos de Revisão *Postmortem* Simplificada e Relatório de Experiências, fazendo recomendações de quando cada um deve ser utilizado. BIRK *et al.* [104] discutem análise *Postmortem* de forma mais genérica.

➤ **Aquisição de Conhecimento**

Aquisição de conhecimento, segundo PROBST *et al.* [68], é a captura de conhecimento de fontes externas à organização, tais como especialistas externos, organizações que têm interesse nas atividades da organização (por exemplo: clientes e fornecedores), além de outras organizações. Nesta atividade, é preciso ter o cuidado de verificar a compatibilidade do conhecimento adquirido com a organização.

Exemplos de serviços/ferramentas de software que apóiam a aquisição de conhecimento são:

- extração, baseada em agentes, de informações de fontes externas disponíveis na Web [57];
- apoio à realização de conferências entre especialistas via Internet, com objetivo pré-estabelecido e protocolo de interação rígido. As contribuições são classificadas, conclusões são elaboradas e os participantes têm a oportunidade de comentar as conclusões [66,105].

➤ **Transformação e Organização de Conhecimento**

Combinando as idéias de alguns autores [10,17,57,58,77,100], a transformação e organização de conhecimento consiste na:

- comparação, análise, combinação e síntese do conhecimento coletado para dar origem a material mais reutilizável (empacotamento), adaptando o conteúdo e o formato para uma situação de uso concreto antecipada;
- revisão do conhecimento obtido por especialistas do domínio e engenheiros de conhecimento para assegurar sua qualidade e potencial de reutilização;

- classificação e indexação apropriada do conhecimento para facilitar sua disseminação, pois os usuários (leitores de conhecimento) esperam achar o conhecimento que eles procuram o mais rápido e fácil quanto possível.

No que se refere aos serviços/ferramentas de software que apóiam a transformação/organização de conhecimento, tem-se

- ferramentas para inclusão de conhecimento segundo a estrutura prevista [9,73,90];
- obtenção semi-automática de índices a partir da extração de informação do respectivo item de conhecimento [58];
- ferramenta para anotação dos itens de conhecimento e estabelecimento das associações necessárias [16].

Em termos de procedimento, NAGEL [17] estabelece: transformação da entrada obtida de acordo com a estrutura de pacote, inserção de informações que descrevem o conteúdo do pacote, associação do pacote aos elementos que permitirão sua recuperação e liberação do pacote para os servidores de conhecimento. Após a liberação, gerentes de projeto são solicitados a comentar os pacotes e outras pessoas também podem fazer seus comentários. Comentários dos comentários são desejados.

➤ **Disseminação de Conhecimento**

A disseminação de conhecimento pode ser ativa, significando que o sistema notifica ou fornece informações que podem interessar ao usuário sem a necessidade de uma solicitação explícita (distribuição de conhecimento), ou pode ser passiva, significando que o usuário deve iniciar a interação com o sistema e especificar o tipo de informação que ele pretende obter (busca de conhecimento) [73].

Inicialmente, os serviços/ferramentas de software voltados para a busca de conhecimento foram considerados extremamente críticos [92]. Atualmente, grande ênfase tem sido dada aos serviços/ferramentas de distribuição de conhecimento. Segundo ABECKER *et al.* [16], os usuários não querem gastar tempo buscando informações, eles precisam de um serviço de distribuição de conhecimento ativo e sensível ao contexto, que saiba o que o usuário está fazendo e explore as informações de forma autônoma.

A maioria dos serviços/ferramentas de software para distribuição de conhecimento é guiada por descrições de processos, enquadrando-se no contexto da Gerência do Conhecimento Orientada a Processo, que será discutida posteriormente. No entanto têm-se, ainda, serviços/ferramentas de software voltados para recomendação de casos de acordo com o perfil do usuário ou similaridade com casos que foram de interesse no passado [58].

No que se refere a serviços/ferramentas de software que apóiam a busca de conhecimento, tem-se:

- busca por atributo [58,73,90];
- buscas por conteúdo ou por exemplo, ambas com recuperação baseada em similaridade [58];
- navegação baseada na estrutura da Memória Organizacional (por exemplo: em áreas de pesquisa, dimensões, conteúdo) [17,58,81,90].

➤ **Manutenção de Conhecimento**

Manutenção de conhecimento se refere à revisão, adaptação e evolução contínua da Memória Organizacional, envolvendo tanto conteúdo quanto estrutura, de forma a lidar com informação incompleta, potencialmente incorreta e em freqüente atualização, e continuamente adicionar valor à Memória Organizacional (combinação das idéias de ABECKER *et al.* [46], BASILI *et al.* [66] e WANGENHEIM *et al.* [58]).

Os exemplos de serviços/ferramentas de software neste contexto são para:

- inclusão de diretrizes para manutenção de conhecimento na própria Base de Experiência, compostas de *trigger*, ações e benefícios esperados, que são usadas para gerar solicitações de mudança para pacotes de experiência que necessitam de manutenção [65];
- identificação, baseada em análise estatística, de termos candidatos para atualização de vocabulário e *thesaurus* [58];
- atualização do perfil do usuário de acordo com as consultas submetidas e casos reutilizados [58];
- coleta de dados sobre a utilidade das experiências armazenadas através de *feedback* do usuário. O usuário indica a utilidade esperada de cada item de experiência retornado em resposta a uma consulta e entrevistas são utilizadas para conhecer a utilidade real dos respectivos itens uma vez que tenham sido utilizados [65].

➤ **Avaliação e Melhoria da Infra-estrutura Técnica**

A avaliação e melhoria da infra-estrutura técnica têm como objetivo promover a sua evolução contínua, de modo a lidar com a complexidade inerente a tal infra-estrutura e com a rapidez com que as Tecnologias da Informação evoluem, acompanhar mudanças nas metas da organização e nos fatores não técnicos envolvidos, e incorporar o aprendizado sobre o que funciona e o que não funciona na organização. Segundo WANGENHEIM *et al.* [58], os mecanismos de acesso e coleta de conhecimento têm que ser revisados e adaptados continuamente.

Os serviços/ferramentas de software que apóiam a avaliação da infra-estrutura técnica têm sido voltados para:

- coleta de *feedback* do usuário através de registro do protocolo de uso [58];
- análise de desvios entre as características de contexto fornecidas, as registradas nos pacotes de experiência e as verificadas na prática [9].

Exemplos de procedimentos propostos com a finalidade de apoiar a avaliação e melhoria da infra-estrutura técnica são:

- um método para melhoria incremental da utilidade da infra-estrutura técnica, que é baseado em: 1) um modelo de uso geral, 2) um modelo de causa e efeito que captura os fatores que influenciam a utilidade percebida pelos usuários, 3) um conjunto de indicadores e 4) uma abordagem de raciocínio baseada em casos que permite tanto interpretação automática quanto análise e interpretação manual. Desta forma, problemas podem ser identificados e a equipe de manutenção pode contatar usuários, melhorar o conteúdo da Memória Organizacional ou os mecanismos de acesso e coleta de conhecimento [74];
- uma abordagem para avaliação da Base de Experiência baseada no método GQM⁸. Metas para a medição são estabelecidas de acordo com as metas de negócio. Para cada meta, entrevistas são feitas com os usuários para identificar critérios de avaliação e suas dependências. Um plano GQM é elaborado, do qual um plano de medição é derivado. Os resultados são interpretados e comparados com as metas em reuniões com os usuários e decisões sobre melhorias são tomadas [107].

2.11 Sistemas

Nesta seção, alguns Sistemas de Gerência do Conhecimento são descritos com o objetivo de exemplificar como a infra-estrutura técnica de Gerência do Conhecimento tem sido implementada em organizações da área de Engenharia de Software.

➤ **Corporate Memory Management System**

Corporate Memory Management System (CMMS) [58] é uma infra-estrutura técnica voltada para organizações de Pesquisa e Desenvolvimento de Software baseada na abordagem da Fábrica de Experiência. A principal meta do sistema é explorar conhecimento/informações existentes na organização, o que inclui conhecimento tácito. A Memória Organizacional é composta de casos, que podem ser de vários tipos e são indexados por indicadores de conteúdo. O conteúdo abrange

⁸ O método GQM (*Goal-Question-Metric*) é uma abordagem para medição orientada a meta em Engenharia de Software [106].

áreas de pesquisa relevantes para a organização e, para cada área de pesquisa, são desenvolvidos vocabulário do domínio, *thesaurus* e um dicionário. A coleta de novos casos está inserida, tanto quanto possível, na execução das atividades dos processos organizacionais, sendo realizada através de formulários *on-line* e podendo ocorrer de forma espontânea, dirigida a evento, planejada ou baseada em recuperação. Esta última representa a captura de conhecimento já existente, mas em formato inadequado. Índices são estabelecidos manualmente ou de forma semi-automática a partir da extração de informações do respectivo caso com base nos vocabulários do domínio. O resultado é revisado e, se necessário, melhorado pelo especialista. O CMMS apóia várias técnicas de busca e recuperação: 1) por navegação dentro uma área de pesquisa, com os casos classificados por tipo, 2) por atributos, 3) por conteúdo, a partir de questões em linguagem natural ou de termos arbitrários e 4) por exemplo, sendo que a recuperação é baseada em similaridade nos dois últimos casos. Quando a busca de conhecimento não fornece uma resposta satisfatória ao usuário, o CMMS permite que ele solicite a resposta diretamente a um especialista via correio eletrônico. A resposta do especialista é automaticamente encaminhada para o usuário e, além disso, dá origem a um novo caso. Em termos de distribuição de conhecimento, o CMMS utiliza três abordagens: filtragem demográfica, que é utilizada para identificar tipos de usuários que se interessam por um certo tópico; métodos baseados em conteúdo, que fazem recomendações baseadas nos casos que foram considerados de interesse para o usuário no passado; e abordagens colaborativas, que recomendam casos que têm sido reutilizados por outros usuários com o mesmo perfil. No CMMS, o esforço requerido para identificação de novos termos para vocabulários e *thesauri* é minimizado pela identificação automática de termos candidatos, baseada, respectivamente, na análise da frequência dos termos nos documentos e na análise de co-ocorrência estatística. Os índices baseados em conteúdo da *profile* do usuário, que permitem a distribuição de conhecimento, são contínua e automaticamente atualizados com base nos índices envolvidos nas consultas do usuário e nos casos reutilizados por ele. No CMMS, o *feedback* do usuário pode ser coletado através de protocolos de uso, em conjunto com os perfis dos usuários e entrevistas.

➤ **Experience Management System**

Experience Management System (EMS) [73,108] é um sistema baseado na abordagem da Fábrica de Experiências que está sendo desenvolvido para uso em uma empresa multinacional de consultoria em Engenharia de Software. O foco do sistema é apoiar a captura, armazenamento, busca e recuperação de experiências de Engenharia de Software que possam ser representadas explicitamente. No EMS, uma

experiência é descrita por um pacote de experiência que é composto de caracterização, relacionamentos e corpo. Uma taxonomia adequada à organização é definida para permitir que os pacotes de experiência sejam fácil e rapidamente encontrados, e pode ser modificada e expandida quando necessário. Cada pacote é associado a um tipo de pacote que estabelece: 1) um conjunto bem definido de atributos para a caracterização dos pacotes baseados em uma taxonomia de classificação, 2) associações que permitem estabelecer relacionamentos entre pacotes, além de 3) elementos para armazenar o conteúdo propriamente dito [73].

Uma das principais características do sistema é a *interface* visual de consulta que permite que o usuário navegue entre as informações armazenadas no repositório através do ajuste de dispositivos de consulta (barras de rolagem e *check box*) e da visualização animada dos resultados na tela do computador. A interface para uma perspectiva selecionada (que corresponde a um tipo de pacote) mostra os pacotes de experiência associados, os atributos da perspectiva e dispositivos para a seleção de valores para estes atributos. O usuário manipula os dispositivos e observa o número de pacotes selecionados em um gráfico tri-dimensional (eixos x, y e cor). Quando um subconjunto pequeno de pacotes é selecionado, o usuário pode, então, examinar cada pacote. SEAMAN *et al.* [108] e MENDONÇA *et al.* [73] destacam o fato da busca ser interativa e controlada pelo usuário, além de permitir a visualização da quantidade de experiências armazenadas e do esquema de classificação utilizado pela organização. Desta forma, o EMS implementa busca por atributo com interface inovadora e amigável e, até o momento, não apóia a distribuição de conhecimento. Outra funcionalidade do sistema permite a submissão de novos pacotes para a Base de Experiências. A funcionalidade utiliza os atributos, associações e elementos definidos nos tipos de pacotes para fornecer os formulários que o usuário deve completar para descrever novos pacotes.

MENDONÇA *et al.* [73] reportam o início do desenvolvimento de uma segunda versão do EMS para uma organização de pesquisa aplicada em Engenharia de Software com o objetivo de adicionar novas interfaces para navegação, busca por palavra-chave, apoio aos diversos papéis existentes em uma Fábrica de Experiência, acesso a arquivos já armazenados em bases de dados corporativas, apoio à anotação de pacotes e avaliação dos mesmos pelos usuários. LINDVALL *et al.* [3] apresentam a experiência de implantação do EMS nas duas organizações já mencionadas e também em um fabricante de interior de automóveis.

➤ ***Integrated Corporate Information Network***

Integrated Corporate Information Network (iCoIN) [85] é uma proposta de Infra-estrutura de Gerência do Conhecimento para um instituto de pesquisa experimental em Engenharia de Software. A arquitetura prevê um conjunto de componentes modulares, os quais são: 1) *Camada de Busca, Agregação e Acesso*, que permite acesso uniforme às fontes de informação, fornece resumos de notícias e fontes de conhecimento relevantes com base no contexto pessoal, funcional e da situação do processo de negócio no qual o usuário está envolvido, além de relatar para a *Camada de Manutenção* o não uso das informações oferecidas e consultas que não retornaram resultados; 2) *Camada de Integração da Informação*, que possibilita a troca de informações entre os diferentes componentes, sendo utilizada principalmente pela *Camada de Busca, Agregação e Acesso* para acessar a informação a ser buscada ou agregada; 3) *Base de Comunidades da Prática*, que atua como uma plataforma para troca e discussão de idéias entre os membros da organização, onde os tópicos de discussão podem ser perguntas que não puderam ser respondidas por outras fontes, além de conteúdo de outras partes do iCoIN; 4) *Portal de Solicitação de Serviços*, que é usado para solicitar serviços internos ou de fornecedores; 5) *Portal de Informação Organizacional*, que fornece acesso uniforme a conhecimento e informações organizacionais, estando disponível um repositório de lições aprendidas baseado na abordagem da Fábrica de Experiências e um repositório de modelos de processo com o objetivo de facilitar a comunicação e o entendimento humano; 6) *Áreas de Trabalho*, que contêm documentos relevantes para a organização ou uma de suas sub-partes (unidades organizacionais, projetos), sendo que seções das *Áreas de Trabalho* dos projetos podem ser acessadas por clientes e parceiros dos projetos; 7) *Suporte ao Planejamento/Administrativo*, que fornece funções usadas para planejamento de atividades dos projetos e para alinhar a organização, como elaboração de cronogramas e registro de competências; 8) *Portal de Informações Externas*, que é uma coleção de fontes WWW relevantes para a organização; 8) *Componente de Manutenção*, que usa o fluxo contínuo de experiências da *Base de Comunidades da Prática* para preencher o repositório de lições aprendidas, fornece resumos dessas experiências e de outros conteúdos para facilitar a definição de agregações para a *Camada de Busca, Agregação e Acesso*, além de monitorar a *Camada de Integração de Informação* para identificar mudanças de situação e ativar atividades de captura de conhecimento. Uma das principais características do iCoIN é o fornecimento de agregações e visões do conteúdo pré-adaptadas de acordo com os principais papéis exercidos na organização, onde o usuário é capaz de instanciar e

ajustar as visões. DECKER e JEDLITSCHKA [85] esclarecem que os primeiros esforços para construção do iCoIN foram dedicados ao repositório de lições aprendidas e ao Subsistema de Gerência da Qualidade que fornece acesso ao repositório de modelos de processo de negócio.

➤ ***Software Engineering Experience Environment***

ALTHOFF *et al.* [74] propõem um ambiente escalável e genérico para o aprendizado e reutilização de todos os tipos de experiência, o *Software Engineering Experience Environment* (SEEE). O armazenamento e a recuperação são baseados em REFSENO, um formalismo de representação para ontologias de engenharia de software [109]. As construções de REFSENO são utilizadas para especificar o esquema da Base de Experiências, que captura o conhecimento conceitual sobre a Base de Experiências, especificando os tipos de experiência que podem ser coletados, como estes tipos de experiência se relacionam e como os tipos de experiência podem ser caracterizados para permitir sua recuperação de forma eficiente. O núcleo da arquitetura do sistema é composto por uma ferramenta de Raciocínio Baseado em Casos, um servidor da Base de Experiências e um *browser* de propósito geral. Ferramentas e sistemas de armazenamento específicos podem ser adicionados quando necessário. Para integrar uma ferramenta específica, é necessário um pequeno módulo de sincronização com o servidor da Base de Experiências. A ferramenta de Raciocínio Baseado em Casos armazena as caracterizações dos itens na Base de Casos e permite pesquisa baseada em similaridade através destas caracterizações. O *browser* de propósito geral permite pesquisa, visualização e manipulação das caracterizações, sendo estas operações tratadas pelo servidor da Base de Experiências e pela ferramenta de Raciocínio Baseado em Casos. O servidor da Base de Experiências sincroniza o acesso à Base de Casos, às bases de dados e aos arquivos e mantém as referências entre as caracterizações e os itens de experiência. Itens específicos de um projeto não precisam ser conhecidos pelo servidor da Base de Experiências.

ALTHOFF *et al.* [74] destacam que o sistema fornece mapas de conhecimento para identificar as fontes de conhecimento tácito (por exemplo: gerente que tem experiência em revisões de projeto), fornece informação relevante para apoiar o planejamento e o acompanhamento do desempenho de projetos (por exemplo: lições aprendidas) e também auxilia a descoberta de conhecimento inesperado.

➤ **Well of experience**

Wox (Well of experience) [101] é um repositório de conhecimento não estruturado que faz parte do conjunto de ferramentas de Gerência do Conhecimento utilizadas por uma empresa de consultoria de médio porte da área de *software*. A ferramenta adota a metáfora das anotações feitas em papel adesivo amarelo. Gerentes e desenvolvedores de *software* são estimulados a registrarem suas anotações pessoais na ferramenta e o uso da ferramenta é promovido através de cartazes em locais movimentados da empresa. *Interfaces* simples são fornecidas para captura e busca das anotações. Cada anotação consiste de assunto, texto descritivo, palavras-chave (sem lista pré-definida), autor e data de submissão. Qualquer pessoa pode adicionar um comentário e atribuir créditos a uma anotação ou enviar a anotação por correio eletrônico para alguém. A *interface* permite busca por palavra-chave, sendo as anotações listadas de acordo com o assunto e os créditos acumulados. Além disso, permite a visualização das anotações ou comentários fornecidos pela própria pessoa, dos créditos recebidos por suas anotações, das 10 últimas anotações adicionadas no repositório, das palavras-chave mais utilizadas e das anotações com mais créditos. A ferramenta tem sido utilizada para resolver problemas técnicos específicos, obter uma idéia geral de uma área de problema específica, evitar a necessidade de explicar uma mesma solução a várias pessoas, ajustar ferramentas para melhoria do trabalho individual, encontrar pessoas com competências específicas na empresa. Dois problemas foram relatados: a pessoa que lê uma anotação nem sempre tem o conhecimento necessário para entendê-la e as pessoas não escrevem sobre todos os tipos de problemas.

➤ **KONTEXT**

KONTEXT (KnOwledge maNagement based on the application conTEXT of software engineering Technologies) [9] é uma ferramenta baseada na abordagem da Fábrica de Experiências para administrar conhecimento sobre tecnologias de Engenharia de Software e seus contextos de aplicação. O conteúdo da Base de Experiência são pacotes de experiência, taxonomias e definições. O pacote de experiência contém a definição da tecnologia e especifica: 1) o processo no qual ela pode ser aplicada, 2) que qualidade de produto de que tipo de produto pode ser obtida usando a tecnologia e 3) em que contexto sua aplicação e, conseqüente, impacto na qualidade podem ser esperados. Um contexto é definido por um conjunto de pares de atributo/valor que definem as características de um projeto de software, sendo um atributo e sua definição referenciados como um fator de contexto. Cada conceito contido em um pacote de experiência é uma parte de uma taxonomia e tem uma

definição associada. Para modelagem dos pacotes de experiência, KONTEXT oferece um componente que permite ao usuário incluir uma nova tecnologia, incluir novo pacote de experiência, modificar um pacote de experiência novo ou selecionado, verificar e validar pacotes de experiência, além de editar taxonomias, definições e fatores de contexto. KONTEXT apóia a verificação automática de pacotes de experiência no que se refere à completude e alguns aspectos básicos de consistência e apóia a validação ao oferecer uma visão bem estruturada dos pacotes para inspeção. Apesar de uma base de experiência em tecnologias de software ser útil para uma série de atividades, o foco inicial do uso da ferramenta foi a seleção de tecnologias durante o planejamento de projetos de software e os programas de melhoria, buscando-se apoiar a tomada de decisão humana. A ferramenta implementa um processo de suporte à decisão que guia o usuário ao longo de múltiplas atividades, o que inclui: caracterização dos objetivos estabelecidos para a aplicação da tecnologia (processo, qualidade e tipo do produto), caracterização do projeto de software ou do programa de melhoria (contexto) segundo questionário adequado à busca por similaridade, seleção de método de suporte à decisão, elaboração da decisão final e justificativa. Ao longo deste processo, o usuário pode acessar informações sobre as tecnologias, explorar diferentes cenários de caracterização comparando resultados, retroceder no processo e iterativamente executar sub-atividades antes de atingir a decisão final. Para apoiar a avaliação empírica da tecnologia selecionada, o registro completo do processo de decisão, incluindo a caracterização do contexto inicial do projeto ou programa de melhoria é armazenado na Base de Experiência, o contexto final pode ser fornecido pelo usuário e armazenado, permitindo que a ferramenta guie o usuário através do processo de análise dos desvios entre as características de contexto esperadas, as reais e as do pacote de experiência. O objetivo da avaliação é possivelmente refinar e atualizar o conteúdo da Base de Experiência. A aplicação da ferramenta em contexto industrial em 1999 forneceu a primeira evidência empírica da validade da abordagem.

➤ **Análise Comparativa dos Sistemas**

Na tabela 2.3, os sistemas apresentados nesta seção são comparados de acordo o apoio fornecido ao mesmo conjunto de atividades adotado na seção 2.10.2. A maioria dos critérios de comparação foi proposta por SNOEK [69], apesar de estarem organizados de forma diferente. As tabelas 2.4 e 2.5 são utilizadas para esclarecer o significado de alguns critérios e dos acrônimos utilizados na comparação.

Tabela 2.3 – Tabela Comparativa dos Sistemas de Gerência do Conhecimento

Critérios de Comparação por Atividade	Sistemas					
	CMMS	EMS	iCoIN	SEEE	Wox	KONTEXT
Identificação	Sim	Não	Sim	Sim	Sim	Não
Desenvolvimento						
Criação	Não	Não	Sim	Não	Não	Sim
Captura	Sim	Sim	Sim	?	Sim	Sim
Aquisição	Não	Não	Sim	Não	Não	Não
Transformação						
Representação	E	E	N,S,E	E	N	E
Classificação e Indexação	Sim	?	?	?	Não	?
Separação entre Itens e Estrutura	Sim	Sim	?	Sim	Não	Sim
Informação Administrativa	?	?	?	?	Sim	?
Situação dos Itens de Conhecimento	?	?	?	?	Sim	?
Composição do Conhecimento	Não	Não	Não	Não	Não	Não
Disseminação						
Tipos de Disseminação	P, A	P	P	P	P	P
Mecanismos de Busca	Sim	Sim	Sim	Sim	Sim	Sim
Mecanismos de Navegação	Sim	Sim	Sim	Sim	?	?
Divulgação da Existência de Novos Itens	Não	Não	?	Não	Sim	Não
Ambiente Pessoal	Sim	Não	Sim	Não	Não	Não
Sensitiva a Contexto	Não	Não	Sim	Não	Não	Sim
Manutenção						
Feedback do Usuário	Sim	Sim	Sim	Não	Sim	Sim
Análise do Feedback do Usuário	Não	?	?	Não	Sim	Sim
Modificação de Itens de Conhecimento	?	?	?	?	?	Sim
Esquecimento de Itens Sem Importância	?	?	?	Não	?	?
Análise da Estrutura da MO	Sim	Não	Sim	Não	Não	Não
Modificação da Estrutura da MO	Sim	Sim	Sim	Não	Não	Sim

Tabela 2.4 – Descrição de Critérios

Critério	Descrição
Classificação e Indexação	Apoio dado à classificação e indexação dos itens de conhecimento
Separação entre Itens e Estrutura	Distinção entre itens de conhecimento e estrutura e/ou entre caracterização do item de conhecimento e o seu conteúdo, o que fornece maior flexibilidade na manutenção do conhecimento
Informação Administrativa	Informações tais como autor e data de registro
Situação dos Itens de Conhecimento	A situação pode, por exemplo, indicar que o item é obsoleto ou que é freqüentemente utilizado
Ambiente Pessoal	Ajuste da disseminação de conhecimento à necessidade do usuário

Tabela 2.5 – Descrição de Acrônimos

Acrônimo	Significado	Acrônimo	Significado
MO	Memória Organizacional	N	Não estruturado
P	Passiva	S	Semi-estruturado
A	Ativa	E	Estruturado
O	Orientado a Processo	?	Nada pôde ser afirmado

2.12 Fatores Não Técnicos

O'LEARY e STUDER [19] relatam consenso de que aplicações de Gerência do Conhecimento devem estar embutidas em estruturas organizacionais que promovam o aprendizado organizacional e que aplicações de sucesso requerem apoio significativo e explícito da gerência, além do envolvimento de todas as pessoas.

A importância de estabelecer na organização uma cultura de apoio à Gerência do Conhecimento é defendida por vários autores [8,49,66,67], porque funcionários podem considerar que irão perder importância quando tiveram o conhecimento que possuem capturado e podem temer que experiências negativas e lições aprendidas baseadas em falhas, ao serem compartilhadas, possam ser utilizadas contra eles. Além disso, funcionários freqüentemente não têm tempo para fornecer e buscar conhecimento e resistem em reutilizar soluções de outras pessoas [8]. Assim, organizações não só devem encorajar, mas também recompensar funcionários que compartilhem seu conhecimento e busquem e utilizem conhecimento de outras pessoas. Algumas das medidas que têm sido utilizadas são:

- reconhecimento público para as pessoas cujas contribuições tenham ajudado a resolver problemas da organização [8];
- instituição de prêmios para as pessoas que regularmente compartilhem informação útil [8] ou para pessoas que mais compartilhem informação em um período de tempo [101];
- acúmulo de pontos para o autor e o revisor de um documento, baseado nos benefícios gerados a cada leitura do documento. Os pontos acumulados podem ser trocados por prêmios ou dinheiro [88];
- vantagem em negociações salariais de acordo com a capacidade de compartilhamento [67,110];
- fortalecimento do sentimento de grupo e de confiança, com o estabelecimento de liderança participativa e participação em atividades comuns no tempo livre [67].

CREIGHTON e ADAMS [111], no entanto, destaca que recompensas individuais diminuem a participação das pessoas em atividades colaborativas, além de destacar a importância do treinamento.

Além disso, JOHANSSON *et al.* [110], numa abordagem voltada para o conhecimento personificado, propõem a criação de uma posição na organização (*Experience Broker*) com o objetivo de funcionar como facilitador para a criação de uma rede interna de indivíduos, possuindo as seguintes atribuições: identificar vários indivíduos que podem fornecer ajuda/experiência aos demais (comunicadores de experiência), facilitar contatos entre os que necessitam de ajuda/experiência e os comunicadores de experiência, esforçar-se para conectar pessoas, mostrar-se sempre disponível e acompanhar os benefícios provenientes das situações de transferência de experiência.

2.13 Lições Aprendidas, Recomendações e Questões

Vários autores relatam lições aprendidas e conseqüentes recomendações para Gerência do Conhecimento. Nesta seção, serão mencionadas apenas as que, de alguma forma, não tenham ainda sido mencionadas.

MEEHAN e RICHARDSON [13] recomendam aprendizado durante a execução das atividades; um sistema de orientação dos funcionários menos experientes, mesmo que informal; envolvimento de funcionários menos experientes ou novos nos processos de software chave; encorajamento explícito e orientação clara para realização de pesquisa externa; documentos fáceis de serem lidos e com identificação do autor; compartilhamento tanto formal quanto informal do conhecimento sobre processos, devendo a gerência organizar e coordenar reuniões para compartilhamento de conhecimento, encorajar funcionários a trabalharem em grupo e reconhecer e premiar empregados que compartilham.

SCHNEIDER *et al.* [77] fazem uma série de recomendações para implementar Gerência do Conhecimento e aprendizado organizacional em empresas de software: todos os funcionários devem entender o que está acontecendo, porque está acontecendo, como isto afeta o seu trabalho e porque a mudança requer tempo; atividades devem ser planejadas; pessoas chave devem ser aconselhadas a ajustar seu próprio comportamento e não só prover recursos; a Gerência de Riscos deve ser aplicada nas iniciativas de melhoria da mesma forma que seria feito em um projeto de desenvolvimento de software; pacotes de experiência devem ser baseados na necessidade antecipada dos usuários, devem ser tão simples quanto possível de serem criados e utilizados; e o usuário deve poder ver, usar e comentar soluções parciais ao longo de várias iterações.

Para DESOUZA [1], uma organização deve claramente definir os incentivos para encorajar o compartilhamento de conhecimento e alocar pessoas a projetos não pelo que elas sabem, mas pelo seu potencial de aprendizado e exploração. Além disso, Sistemas de Gerência do Conhecimento devem encorajar o diálogo entre indivíduos e não somente apontar para repositórios, devendo-se reconhecer que as Tecnologias da Informação não são o único meio de promover o conhecimento.

MURCH [11] defende que a Gerência do Conhecimento deve ser voltada para a aplicação prática do conhecimento e deve mostrar potencial para agilizar ciclos de desenvolvimento e gerar lucro.

Por fim, KEMP *et al.* [47] destacam, como fator crítico de sucesso, o estabelecimento de expectativas realistas, pois Gerência do Conhecimento é uma atividade complexa.

KING *et al.* [112] conduziram um estudo em 2000 com profissionais envolvidos com Gerência do Conhecimento para identificar as principais questões da área. Foram identificadas 20 questões, sendo as 10 mais freqüentes: como utilizar a Gerência do Conhecimento para obter vantagem estratégica; como obter suporte da gerência de mais alto nível; como manter o conhecimento organizacional atualizado; como motivar indivíduos a contribuírem com conhecimento; como identificar conhecimento organizacional que deve ser capturado; como avaliar os custos financeiros e os benefícios da Gerência do Conhecimento; como verificar a eficácia, a legitimidade e a relevância das contribuições; como melhor projetar e desenvolver um sistema de Gerência do Conhecimento; como sustentar o progresso da Gerência do Conhecimento na organização e como garantir a segurança do conhecimento armazenado.

2.14 Conclusão

Este capítulo teve como objetivo descrever o estado da arte da Gerência do Conhecimento, de forma a fornecer subsídios e idéias para a definição e construção de Ambientes de Desenvolvimento de Software Orientados a Organização.

Capítulo 3 - Gerência do Conhecimento Orientada a Processos

3.1 Introdução

Em 2001, pesquisas indicavam uma tendência em direção à integração da Gerência do Conhecimento com a Modelagem de Processos de Negócio [71]. Um seminário realizado na Alemanha chegou ao consenso de que a Gerência do Conhecimento deve ser freqüentemente embutida em processos e que estes devem ser re-projetados para acomodar a Gerência do Conhecimento [19]. No entanto, antes de se tornar tendência, esta abordagem já vinha sendo adotada [46,82] e mesmo defendida [10]. No que se refere a processos de software, alguns trabalhos [6,113] foram mencionados por HENNINGER [96], que destacou a necessidade de considerar a natureza da atividade de desenvolvimento de software e, assim, fornecer apoio a desvios, estabelecimento de planos alternativos e atualização de planos de acordo com a situação. Este capítulo discute a abordagem no contexto da Engenharia de Software, apresentando alguns sistemas e relatos de experiência.

3.2 Integração de Gerência do Conhecimento com Modelagem de Processos

Segundo JABLONSKI *et al.* [90], a integração de Gerência do Conhecimento com Modelagem de Processos de Negócio consiste em utilizar os processos de negócio como uma dimensão para a organização do repositório de conhecimento. Como o repositório de conhecimento necessita de dimensões adicionais de organização para tratar conhecimento que não pode ser diretamente associado a processos de negócio, a abordagem não pode ser considerada dependente de processo, mas orientada a processo, dando origem ao termo Gerência do Conhecimento Orientada a Processos.

Para MAIER e REMUS [18], a Gerência do Conhecimento Orientada a Processos tem como objetivo fornecer, ao longo dos processos de negócio, conhecimento relacionado às tarefas para os funcionários da organização. No entanto, a orientação a processo pode contemplar desde um único processo da organização a todos os processos de negócio relevantes, sendo que começar com um único processo pode trazer vantagens em termos de aceitação. Os autores também destacam que identificar os tipos de processos que são candidatos promissores à

Gerência do Conhecimento Orientada a Processos é uma questão fortemente relacionada à identificação dos processos de negócio da organização que são intensos em conhecimento, sendo que, dentro deste grupo, ainda é possível distinguir entre processos simples e altamente complexos, e entre processos básicos, de gerência e de serviço.

3.3 Modelagem de Processos

Modelos de processo descrevem tarefas genéricas, sua decomposição em sub-tarefas e o fluxo de informação entre as tarefas [6], podendo ser elaborados com diferentes propósitos: facilitar a comunicação e o entendimento humano, apoiar a melhoria dos processos, apoiar a gerência dos processos, automatizar o fornecimento de orientações e automatizar a execução dos processos. Quanto ao uso, modelos de processo podem ser ainda descritivos ou prescritivos [85]. Neste sentido, JABLONSKI *et al.* [90] acrescentam que a Modelagem de Processos tem várias facetas: descreve como processos de negócio são executados de forma a fornecer modelos para futuras execuções; prescreve como os recursos são utilizados de forma otimizada para atingir o objetivo do processo, além de não ser limitada por fronteiras organizacionais ou técnicas dentro de uma organização, englobando todos os componentes relevantes ao conteúdo independente das unidades aos quais pertencem.

3.4 Justificativas e Requisitos

Várias são as justificativas para a Gerência do Conhecimento Orientada a Processos:

- modelos de processo representam conhecimento [generalizado de 6];
- indivíduos precisam entender os papéis que desempenham nos processos [5];
- a organização precisa entender seus processos para ser capaz de melhorá-los e precisa torná-los disponíveis para seus funcionários para direcionamento [5,66];
- pessoas não precisam de conhecimento por si mesmo, mas conhecimento para executar tarefas específicas [94];
- processos são entidades naturais para gerentes e membros de equipes de projetos de *software*, ou seja, eles estão acostumados a pensar em processos [94];
- processos fornecem parte do contexto que é importante para a disseminação de conhecimento de acordo os objetivos dos usuários e para a interpretação e construção de conhecimento relativo aos próprios processos [18].

No entanto, segundo MAIER e REMUS [18], esta abordagem depende fortemente das seguintes condições: 1) os processos de negócio terem sido

modelados e descritos e serem, portanto, conhecidos pelos funcionários da organização; 2) as atividades de gerência de processo (por exemplo: reengenharia de processos de negócio, melhoria de processos de negócio) já terem sido conduzidas na organização.

3.5 Abordagens para Integração de Gerência do Conhecimento com Modelagem de Processo

ALLWEYER [10] propõe que os modelos de processo sejam usados não só para descrever e analisar o processamento de conhecimento, mas também para permitir o acesso ao conhecimento sempre que isto for necessário, fornecendo uma *interface* intuitiva para buscar conhecimento organizacional.

HOLZ *et al.* [94] propõem a associação explícita das tarefas com o conhecimento necessário a elas, permitindo que uma máquina de processo guie os usuários em suas tarefas diárias, interpretando as descrições explícitas de processo e fornecendo acesso ao conhecimento necessário.

Algumas idéias de ABECKER *et al.* [16] para integração de Gerência do Conhecimento com Modelagem de Processo são:

- quando usados para organizar o conteúdo da Memória Organizacional, modelos de processos de negócio podem ser usados para navegação manual;
- é possível especificar a necessidade de informações/conhecimentos para cada atividade de um processo executado por uma máquina de *workflow*. Quando uma atividade específica é iniciada, uma consulta à Memória Organizacional, de acordo com a necessidade anteriormente especificada, pode ser automaticamente feita e ativamente oferecer o resultado para o usuário;
- a especificação da necessidade de informações/conhecimentos pode ser parametrizada para considerar as necessidades específicas da instância, de forma a permitir recuperação de informações/conhecimentos específicos para o contexto de cada instância;
- se o Sistema de Gerência do Conhecimento conhece o contexto de criação de um documento durante a execução do *workflow*, esta informação pode ser arquivada com o documento e ser utilizada para avaliar a qualidade do conhecimento e para melhorar a recuperação do documento em situações similares;
- o usuário deve ter possibilidade de facilmente fazer comentários, iniciar discussões e enviar e-mails para autores e gerentes de conhecimento caso surja alguma crítica ao material recebido.

No entanto, processos de *software* são processos intensos em conhecimento e, conseqüentemente, tendem a ser caracterizados por mudanças dinâmicas e padrões de colaboração e comunicação altamente individuais e *ad hoc*, o que os torna difíceis de serem planejados em detalhes com antecedência. ABECKER *et al.* [16] propõem duas abordagens para tratar a questão. Uma, que consideram mais simples e dizem preservar a maioria dos benefícios da integração, é modelar o processo sem muitos detalhes e embutir sub-tarefas intensas em conhecimento como caixas pretas. A outra abordagem é, abaixo do nível da granularidade que pode ser pré-estabelecida, compor *workflows* específicos para determinados casos a partir de fragmentos de processos, adaptar o *workflow* em tempo de execução e evoluir o repositório de fragmentos através da reflexão sobre a ação.

3.6 Sistemas e Relatos de Experiência

Os sistemas e relatos de experiência apresentados nesta seção se referem, principalmente, à Gerência do Conhecimento Orientada a Processos em organizações que desenvolvem *software*. No entanto, alguns exemplos voltados para processos intensos em conhecimento são também fornecidos, uma vez que, como já mencionado, processos de *software* são intensos em conhecimento.

➤ **Software Experience Center**

O projeto SEC (*Software Experience Center*) [75] é uma proposta alternativa à abordagem clássica da Fábrica de Experiência que é voltada para dados qualitativos e modelos de processo, e que apóia desde o desenvolvimento de experiências até a disponibilização das mesmas para a tarefa de *software* sendo executada [77]. As experiências são disseminadas através das descrições de processo disponíveis na Intranet. Uma descrição de um processo na Intranet fornece acesso a materiais de treinamento, experiências, *checklists*, perguntas freqüentes e endereços de *e-mails* de especialistas. SCHNEIDER *et al.* [77] definem uma base de experiência como o armazenamento persistente de experiências organizadas em torno de um processo de *software*.

➤ **T-Nova Development Center South West**

NAGEL [17] relata a experiência de uma unidade de desenvolvimento de *software* para a área de telecomunicação, na qual a Gerência do Conhecimento é baseada, apoiada e guiada pelos processos. Todos os processos do modelo de processo de negócio são cobertos pelo Sistema de Gerência do Conhecimento (principalmente os processos de consultoria e engenharia de *software*) e as descrições

de processo são efetivamente utilizadas no trabalho. Os processos de *software* padrão, por exemplo, são adaptados para a obtenção dos processos de *software* específicos. A abordagem consiste em utilizar processos para organizar as experiências da organização, ou seja, pacotes de experiência são associados a sub-processos ou atividades. NAGEL [17] descreve tanto a estrutura dos pacotes de experiência quanto dos processos de negócio. As vantagens são: 1) os membros da organização obtêm os processos e atividades que eles têm que executar (definição do que fazer) juntamente com experiências (explicação de como pode ser feito), tais como exemplos e soluções; 2) experiências similares de departamentos/equipes diferentes podem ser detectadas através da análise dos pacotes de experiência anexados a uma atividade e podem ser utilizadas para melhorar os processos ou uma parte deles; 3) quando alguém tem uma dúvida, ele/ela identifica a atividade ou sub-processo e verifica se há experiências anexadas. As experiências são obtidas através de coleta espontânea; de coleta nas revisões de projeto, em que as experiências coletadas são classificadas em melhores práticas, não práticas e informações; e de coleta nas avaliações de projeto, em que são coletadas apenas melhores práticas, pois só departamentos e equipes publicam não práticas e apenas quando desejarem. A ferramenta para coleta espontânea não requer que o usuário estruture a contribuição, cabendo ao engenheiro do conhecimento transformar e incorporar a contribuição na base de conhecimento. Muito tem sido incluído na base de conhecimento através de coleta espontânea e um grande número de revisões e avaliações de projeto é escrito anualmente. A equipe de consultores e engenheiros de conhecimento é responsável pela qualidade dos processos utilizados pelas equipes de projeto e departamentos, assessora estas equipes e é, também, responsável pelo conteúdo da base de conhecimento. Quando um projeto é iniciado, os engenheiros de conhecimento analisam as características do projeto em conjunto com os gerentes do projeto, procuram pacotes de experiência baseados nestas características e dão consultoria à equipe do projeto. Equipes de projeto usam a base de conhecimento através do processo padrão, navegando pelos pacotes de experiência associados ao processo, ou buscam pacotes de experiência interessantes para o projeto através do acesso disponível na intranet, utilizando, para isto, as diferentes listas de pacotes ou a máquina de busca. A abordagem foi considerada pragmática e bem-sucedida e um dos fatores de sucesso foi o fato da organização já ter o modelo de processo de negócio bem definido.

➤ ***Building an Organizational Repository of Experiences***

BORE (*Building an Organizational Repository of Experiences*) é um protótipo exploratório surgido em 1997 que evoluiu de simples repositório de experiências para uma ferramenta que combina memória organizacional com processo de *software*, sendo orientada por uma metodologia que define como as informações existentes na memória organizacional são recuperadas para apoiar novos projetos e como novo conhecimento é capturado para atender às necessidades emergentes dos esforços de desenvolvimento de *software* [114]. A Memória Organizacional é dividida em domínios que representam núcleos de conhecimento independentes, onde cada domínio é composto de casos e regras do domínio. Casos do domínio definem as atividades que têm sido consideradas úteis para projetos do domínio e são estruturados em uma hierarquia de decomposição de trabalho. Regras do domínio são utilizadas para adaptar o processo genérico definido para o domínio de acordo com as necessidades específicas dos diferentes projetos e consistem de um conjunto de pré-condições e ações, onde as pré-condições são pares de pergunta e resposta. Quando um projeto é criado no BORE, ele é associado a um único domínio e os casos e regras do domínio são utilizados para definir as atividades do projeto, sendo um novo caso instanciado para cada uma das atividades do projeto. Como um domínio pode definir mais de uma trilha de atividades para acomodar os diferentes tipos de projeto conduzidos no domínio, um sistema de apoio à decisão apresenta as opções associadas às atividades na forma de perguntas e possíveis respostas. Uma vez selecionadas as opções que melhor representam as características do projeto, as regras que tiverem todas as pré-condições satisfeitas são ativadas e o conjunto definido de ações é executado, selecionando atividades para o projeto dentre as definidas pelo domínio. Desta forma, as atividades atribuídas ao projeto são expandidas na medida em que o conhecimento necessário para seleção das opções é obtido. BORE exibe as atividades do projeto como uma hierarquia de casos, que são compostos de descrição da atividade, informações específicas do projeto acumuladas na execução da atividade, documentos anexados à atividade, além de casos relacionados, proprietário e situação. Desta forma, casos de projeto são criados a partir da experiência acumulada no domínio, são utilizados para disseminação de experiências de projetos e podem ser usados como um *checklist* dinâmico para captura de experiências. O usuário também pode facilmente iniciar um diálogo “O que – se” ao mudar as respostas dadas ao sistema de apoio à decisão e pode obter informações sobre cada uma das questões que precisa responder. Quando um projeto necessita desviar-se do processo genérico estabelecido pelo domínio, o processo resultante é analisado,

aprovado e documentado para fornecer uma trilha para projetos com características semelhantes, o que pode envolver criar regras, casos e/ou associar novas opções aos casos do domínio. As regras do domínio e as questões e possíveis respostas (utilizadas para estabelecer pré-condições e para a caracterização do projeto) são criadas e editadas através do *Rules Manager*. Os casos do domínio são criados e editados através da mesma *interface* utilizada para editar e acessar a hierarquia de casos do projeto (*Case Manager*), que também permite a associação de questões para caracterização dos projetos [86,114].

HENNINGER e SCHLABACH [114] explicam que casos e regras de domínio podem ser utilizados para definir processos de *software* em qualquer nível de detalhe, apresentando um exemplo de aplicação do BORE para a definição e uso de metodologias de desenvolvimento padrão, detalhado em [86], e outro que mostra como um conjunto de modelos parametrizados pode ser utilizado para definir arquivos de configuração de banco de dados e *scripts* de base de dados.

➤ **Subsistema de Gerência da Qualidade do iCoIN**

Apesar do iCoIN [85], descrito no capítulo anterior, não enfatizar a integração entre Modelagem de Processo e Gerência do Conhecimento, o Subsistema de Gerência da Qualidade, que cobre os modelos de processo de negócio, foi desenvolvido com o objetivo de prover uma integração fraca dos serviços oferecidos pelos vários componentes, exatamente por descrever o uso dos repositórios de informação e conhecimento e fornecer ponteiros para estes repositórios. A técnica de modelagem é texto estruturado e processos complexos são estruturados em uma hierarquia de super e sub-processos, na qual os objetos são Descrição de Processo, Descrição de Papel, Descrição de Agente, Descrição de Produto e *Overview*. A Descrição de Processo contém, entre outras seções, uma seção de Modelos, Exemplos e Informação Adicional, que lista produtos relacionados ao processo, e uma seção de Diretrizes, que dá sugestões de como executar um processo (tais como o que deve ser feito e o que não deve ser feito), fornece resposta a questões freqüentes sobre o processo, além de desvios do processo freqüentemente utilizados. Uma Descrição de Processo está associada a Descrições de Papel, que informam quem pode desempenhar um papel no processo e quem pode substituir se o respectivo agente não estiver disponível, atuando como “páginas amarelas” do iCoIN. A Descrição de Produto é composta de representação e uso, onde representação indica como o produto pode ser acessado e uso indica se o produto é um modelo, exemplo de documento ou informação adicional para a descrição do processo ou papel. Os objetos *Overview* apóiam a navegação ao estruturar Descrições de Processo ou de

Papéis de acordo com algum critério. Segundo DECKER e JEDLITSCHKA [85], as descrições de processos tem uma função particular na integração das atividades de Gerência do Conhecimento, definindo o ponto na execução dos processos de negócio onde serviços como captura de conhecimento ou acesso às fontes devem ser executados e descrevendo as próprias atividades de Gerência do Conhecimento.

➤ ***DElivery of Context-sensitive ORganizational knowledge***

O projeto DECOR (*DElivery of Context-sensitive ORganizational knowledge*) [16] tem como objetivo desenvolver, testar e consolidar abordagens para fornecer: (i) arquivos de conhecimento organizados em torno de representações formais de processos de negócio para facilitar navegação e acesso, (ii) serviços de distribuição de informação que ofereçam ao usuário, de maneira sensível ao contexto, informação útil do arquivo de conhecimento e (iii) métodos para analisar a organização sob a perspectiva do conhecimento, que são métodos requeridos para apoiar os dois primeiros itens. A abordagem utilizada no projeto DECOR consiste de um sistema de gerenciamento de *workflow*, que sabe as tarefas específicas que estão sendo executadas pelos usuários em um dado momento, e de um assistente de informação, que observa a execução do *workflow* e interpreta as necessidades de informação modeladas de forma a oferecer apoio ativo através de conhecimento proveniente do arquivo orientado a processo. Além disto, mantém uma noção do contexto de recuperação da informação utilizando variáveis de fluxo de informação, o que permite consultas mais precisas. Este contexto também pode ser usado para descrever o contexto de criação de um dado item. O cenário acima é baseado em um número de passos de análise e modelagem não triviais que são necessários para obter: 1) os modelos dos processos de negócio (em termos de uma ontologia) e as ontologias de domínio para organização do conhecimento e descrição do conteúdo; 2) os *workflows* fracamente estruturados para processos de negócio intensos em conhecimento; juntamente com 3) o fluxo de informação e as necessidades de informação para enriquecimento dos *workflows*. Sendo assim, o projeto DECOR desenvolveu uma metodologia para sistematicamente executar projetos de Gerência do Conhecimento Orientada a Processos de Negócio e estabeleceu uma infra-estrutura composta de: 1) uma ferramenta de modelagem de conhecimento de negócio que apóia, de maneira integrada, todas as atividades de modelagem; 2) *Sistema de Arquivo Básico* que constitui a Memória Organizacional, armazenando itens de conhecimento anotados com meta-dados e associados a outros itens de conhecimento, onde os meta-dados são representados em termos das ontologias e os itens de conhecimento são documentos, ponteiros para documentos ou ponteiros para conhecimento tácito (por

exemplo, para uma *homepage* em um sistema de páginas amarelas com resumos de competências); 3) *Interface de Anotação*, que é uma ferramenta de *software* para facilmente anexar categorias semânticas para os itens de conhecimento, de forma a incluí-los no *Sistema de Arquivo Básico*, indexá-los e estabelecer as associações necessárias; 4) uma ferramenta de *workflow* fracamente estruturado para apoiar a modelagem e a execução de *workflow* flexível e adaptativo; 5) agentes de conhecimento conscientes do contexto, que têm o propósito de cooperar com a máquina de *workflow* e, de acordo com as necessidades de informação modeladas, ativamente oferecer informação do *Sistema de Arquivo Básico* para o usuário responsável por uma certa atividade.

➤ **Process Oriented Knowledge Management System**

JABLONSKI *et al.* [90] apresentam um protótipo de Sistema de Gerência do Conhecimento Orientada a Processo (*Process Oriented Knowledge Management System*), cujo conteúdo é estruturado em termos de armazenadores de conhecimento associados a dimensões que melhor caracterizam seu conteúdo e etiquetados por atributos que os classificam de forma independente das dimensões. Uma dimensão composta pelos processos de negócio é automaticamente fornecida, mas outras dimensões podem ser definidas, devendo, de forma ideal, representar estruturas físicas ou lógicas da organização. Além disso, dimensões podem ser decompostas hierarquicamente e esta decomposição pode ser estendida posteriormente. Os componentes do sistema são: 1) *Gerenciador de Know How*, componente principal do sistema que atua como servidor disponibilizando armazenadores de conhecimento; 2) *Interface de Processo*, que é usada para alimentar o *Gerenciador de Know How* com processos e que pode ser facilmente adaptada para conectar-se com quaisquer ferramentas de modelagem de processo, 3) *Interface de Armazenador*, que permite trazer para a base de conhecimento os armazenadores de conhecimento, seja fisicamente ou por referência, 4) *Base de Dados*, que armazena dimensões, armazenadores de conhecimento e atributos, compondo, junto com o *Gerenciador de Know How*, a Base de Conhecimento Orientada a Processo; e 5) *Interface WWW*, que fornece acesso à Base de Conhecimento Orientada a Processo e permite que dimensões sejam definidas em adição à dimensão de processo, que é tratada separadamente pela *Interface de Processo*. O usuário pode realizar uma pesquisa baseada nos atributos associados aos armazenadores de conhecimento, pode navegar através das dimensões do sistema e pode acessar documentos que estão associados à combinação das seleções correntes nas múltiplas dimensões. Para inserir um armazenador de conhecimento na base de conhecimento, o usuário

primeiro seleciona valores que melhor caracterizam o conteúdo do armazenador de conhecimento de acordo com as dimensões relevantes, depois lê o armazenador de conhecimento e, então, o armazenador de conhecimento pode ser associado com os atributos que o caracterizam adicionalmente. O sistema tem sido usado em uma grande indústria de carros alemã e os usuários reportaram maior familiaridade com a base de conhecimento e maior disposição para usá-la devido à introdução da dimensão de processo na base de conhecimento. Além disso, os usuários apreciaram a melhoria conceitual em relação à modelagem de processo pura, desde que informação adicional sobre os processos pode ser armazenada na base de conhecimento.

➤ **DFKI KnowMore**

DFKI KnowMore, proposto por ABECKER *et al.* [46,82], tem o objetivo de atuar como um assistente inteligente que acompanha a execução das tarefas e apresenta informações e conhecimento relevantes, que ajudam o funcionário a realizar melhor e mais efetivamente o seu trabalho. O modelo do sistema é composto de três camadas: nível de aplicação, nível de descrição e nível de objeto.

O nível de objeto é caracterizado por várias fontes de informação e conhecimento, que utilizam desde representações formais, que podem ser lidas por máquina, até representações informais, voltadas para a leitura humana.

O nível de descrição é baseado em lógica e rico em conhecimento, de modo a permitir acesso uniforme e inteligente às diversas fontes de informação e de conhecimento a partir de necessidades específicas. Neste nível de modelagem de informação, todo item de conhecimento e informação é descrito por um conjunto de atributos representando o meta-modelo da informação, o conteúdo da informação e o contexto de criação e aplicação. O meta-modelo da informação descreve os diferentes tipos de fontes de informação com respectiva estrutura, acesso e propriedades de formato, utilizando para isto o vocabulário especificado na Ontologia da Informação. A Ontologia da Informação contém conceitos e atributos genéricos que independem de tipos de informação e também inclui conceitos e atributos para certos tipos específicos de informação. O conteúdo da informação é descrito através da Ontologia do Domínio e os contextos de criação e de utilização dos itens de informação são descritos através de uma Ontologia de Empresa. A utilização de uma Ontologia do Domínio para descrever o conteúdo permite inferências formais sobre o conhecimento representado e a formulação de heurísticas de recuperação que explorem a estrutura do domínio. A ontologia de empresa permite expressar o contexto da informação em termos da estrutura e dos modelos de processos organizacionais.

Por fim, o nível de aplicação associa o modelo de informação e a situação de aplicação concreta. Parâmetros do contexto atual de trabalho são mapeados dentro de expressões, resultando em consultas e inserções apropriadas. Quando um funcionário reconhece a necessidade de uma informação ao longo do seu trabalho, é derivada uma consulta para a Memória Organizacional, que é instanciada e restringida o máximo possível de acordo com o contexto de trabalho atual. Quando uma nova informação é inserida na Memória Organizacional, isto é feito de forma enriquecida de contexto, de modo que, em processos de recuperação subseqüentes, a situação da consulta possa ser comparada com a situação de criação para estimativa da relevância específica para o contexto. Para desenvolver uma abordagem geral para avaliar essa relevância, foram selecionados a modelagem de processos de negócio e o uso de um sistema de gerência de *workflow*. Durante a execução do *workflow*, as instâncias das variáveis formam o contexto da tarefa, fornecendo, para a atividade atual, a informação necessária sobre o ambiente. Uma representação para tarefas intensas em conhecimento (tarefas de conhecimento) foi elaborada para estender o formalismo da modelagem de processos de negócio. Esta representação caracteriza uma tarefa de conhecimento através de nome da informação necessária, conjunto de pré-condições que orientam quando a necessidade da informação deve ser considerada, uma consulta parametrizada para recuperação da informação, uma declaração das metas para as quais a informação pode contribuir, um conjunto de regras pós-processadas que influenciam a apresentação de informação não formal ou guiam o processamento de dados formais. A ativação de uma tarefa de conhecimento, controlada pelo serviço de execução de *workflow*, pode começar a partir de uma aplicação convencional ou através de um agente de conhecimento.

➤ ***Customizable Organizational Resources Environment with Knowledge Management***

CORE-KM (*Customizable Organizational Resources Environment with Knowledge Management*) é um ambiente customizável, que está sendo definido e construído para apoiar a Gerência do Conhecimento em diferentes organizações [115]. Reconhecendo que o ambiente que gerencia o conhecimento de uma organização deve atender às necessidades específicas da organização, o CORE-KM apóia a definição, customização e execução de ambientes de gerência do conhecimento específicos para as diferentes organizações.

A infra-estrutura do CORE-KM é composta por um conjunto de funcionalidades, que representam as atividades apoiadas pelo ambiente, e por uma memória organizacional central. As atividades apoiadas são a identificação das necessidades

de conhecimento da organização e as atividades de captura, criação, disseminação, utilização, valoração e manutenção do conhecimento. Dessas atividades, apenas a primeira é específica do ambiente customizável, sendo as demais também apoiadas nos ambientes customizados.

A identificação das necessidades de conhecimento ocorre através de entrevistas e questionários e envolve a identificação dos processos da organização intencionalmente em conhecimento. A captura de conhecimento ocorre segundo o processo definido por MONTONI [31], permitindo a captura de conhecimento ao longo dos processos organizacionais ou de forma independente dos mesmos. A disseminação de conhecimento ocorre ao longo dos processos organizacionais, na medida em que é necessário para a execução de uma atividade. A valoração será realizada a partir de um repositório de métricas fornecido com o ambiente customizado, mas passível de evolução. A manutenção do conhecimento implicará na limpeza periódica das bases que compõem a memória organizacional com exclusão de conhecimento não utilizado e considerado não relevante. Tanto a criação de conhecimento quanto a utilização de conhecimento foram definidas como atividades que serão executadas de acordo com processos específicos a serem definidos para cada organização, sendo que a criação de conhecimento será apoiada apenas em ambientes customizados para organizações que executam atividades de pesquisa.

A memória organizacional no CORE-KM contém um conjunto de bases de conhecimento vazias que representam a infra-estrutura de conhecimento. Cada ambiente customizado tem uma memória organizacional, composta das bases selecionadas da infra-estrutura de conhecimento e das bases pertencentes às ferramentas específicas que tenham sido desenvolvidas para a organização. Como proposto por GALOTTA *et al.* [115], a memória organizacional dos ambientes customizados pode incluir ontologias, uma biblioteca digital, bases de dados, bem como bases de currículos, propostas, diretrizes, melhores práticas e lições aprendidas.

Para auxiliar o engenheiro do conhecimento na definição e customização do ambiente de gerência do conhecimento, o CORE-KM fornece uma ferramenta que guia o engenheiro do conhecimento na execução do processo de customização (CORE-KM *Customizer*). A ferramenta possui uma representação explícita do processo e permite que, a cada atividade do processo, o engenheiro do conhecimento tenha acesso ao conhecimento relevante para a sua execução e registre sugestões relacionadas à atividade. Outras ferramentas estão sendo definidas e construídas.

Dois customizações já foram realizadas a partir do CORE-KM, uma para uma empresa de desenvolvimento de software, que deu origem ao ambiente customizado *TecKnowledge*, e outra para a FBC, organização citada no capítulo 1, que deu origem

ao ambiente customizado *CardioKnowledge*. Para cada um desses ambientes customizados, foi desenvolvido uma ferramenta específica para o ambiente. A *Software Supply Manager* apóia o processo de fornecimento no ambiente *TecKnowledge* e a *CardioResearch* apóia os processos de participação em estudos multicêntricos internacionais e os processos de assistência e pesquisa em ambulatórios hospitalares.

➤ Análise Comparativa dos Sistemas

Na tabela 3.1, os sistemas apresentados nesta seção são comparados de acordo o apoio fornecido ao conjunto de atividades proposto na seção 2.10.2. Para a comparação, são utilizados os mesmos critérios adotados na comparação dos sistemas descritos no capítulo anterior. Para facilitar a leitura, são novamente fornecidas as tabelas para esclarecimento do significado dos acrônimos e de alguns critérios (tabelas 3.2 e 3.3).

Tabela 3.1 – Tabela Comparativa dos Sistemas de Gerência do Conhecimento

Critérios de Comparação por Atividade	Sistemas					
	BORE	Q-iCoIN	DECOR	POKMS	DFKI	CORE-KM
Identificação	Não	Sim	Sim	Não	Não	Sim
Desenvolvimento						
Criação	Sim	Não	Não	Não	Não	Sim
Captura	Sim	?	Sim	Sim	Sim	Sim
Aquisição	Não	Não	Não	Não	Não	Não
Transformação						
Representação	E	S,E	N,S,E	N,S,E	N,S,E	E
Classificação e Indexação	Não	?	Sim	?	?	?
Separação entre Itens e Estrutura	Não	Sim	Sim	Sim	Sim	?
Informação Administrativa	Sim	?	?	?	Sim	?
Situação dos Itens de Conhecimento	Sim	Sim	?	?	?	?
Composição do Conhecimento	Não	Não	Não	Não	Não	?
Disseminação						
Tipos de Disseminação	O	O	O, A	O	O, A	O
Mecanismos de Busca	?	?	Sim	Sim	Sim	?
Mecanismos de Navegação	?	Sim	Sim	Sim	?	?
Divulgação da Existência de Novos Itens	Não	Não	Não	Não	Não	?
Ambiente Pessoal	Não	Não	Não	Não	Não	?
Sensitiva a Contexto	Sim	Sim	Sim	?	Sim	Sim
Manutenção						
Feedback do Usuário	Sim	?	?	?	?	Sim
Análise do Feedback do Usuário	Sim	?	?	?	?	?
Modificação de Itens de Conhecimento	Sim	?	?	?	?	?
Esquecimento de Itens Sem Importância	?	?	?	?	?	Sim
Análise da Estrutura da MO	?	?	?	?	?	?
Modificação da Estrutura da MO	?	?	?	Sim	?	?

Tabela 3.2 – Descrição de Critérios

Critério	Descrição
Classificação e Indexação	Apoio dado à classificação e indexação dos itens de conhecimento
Separação entre Itens e Estrutura	Distinção entre itens de conhecimento e estrutura e/ou entre caracterização do item de conhecimento e o seu conteúdo, o que fornece maior flexibilidade na manutenção do conhecimento
Informação Administrativa	Informações tais como autor e data de registro
Situação dos Itens de Conhecimento	A situação pode, por exemplo, indicar que o item é obsoleto ou que é freqüentemente utilizado
Ambiente Pessoal	Ajuste da disseminação de conhecimento à necessidade do usuário

Tabela 3.3 – Descrição de Acrônimos

Acrônimo	Significado	Acrônimo	Significado
MO	Memória Organizacional	N	Não estruturado
P	Passiva	S	Semi-estruturado
A	Ativa	E	Estruturado
O	Orientado a Processo	?	Nada pôde ser afirmado

3.7 Conclusão

Este capítulo abordou a integração de Gerência do Conhecimento com a Modelagem de Processos de Negócio, dando origem à Gerência do Conhecimento Orientada a Processos. A abordagem foi descrita por ser a abordagem que será utilizada para a introdução de Gerência do Conhecimento nos Ambientes de Desenvolvimento de Software Orientados a Organização.

Capítulo 4 - Ambientes de Desenvolvimento de Software

4.1 Introdução

Ambientes de Desenvolvimento de Software (ADS) têm evoluído ao longo do tempo para fornecer apoio mais amplo e efetivo aos desenvolvedores de software, de forma que metas como aumento da produtividade, melhoria da qualidade, diminuição de custos e, mais recentemente, diminuição do tempo para introdução no mercado possam ser alcançadas. Neste contexto, surgiram os Ambientes de Desenvolvimento de Software Centrados em Processo, os Ambientes de Desenvolvimento de Software Orientados a Domínio (ADSOD) e, mais recentemente, a idéia de integrar Gerência do Conhecimento em ADS.

Este capítulo apresenta, de forma sucinta, um histórico da evolução dos ADS, concentrando-se, pela importância para a fundamentação deste trabalho, nos ADS Centrados em Processo, na descrição de um modelo para construção de ADSOD, incluindo um exemplo de sua implementação, e na descrição de dois ADS Centrados em Processo que apóiam a Gerência do Conhecimento.

4.2 Histórico

Ambientes de Desenvolvimento de Software (ADS) são sistemas computacionais que apóiam o desenvolvimento, reparo e melhoria de produtos de software, e o controle e gerenciamento destas atividades [116]. Para isto, são compostos de um ou mais repositórios, que armazenam todas as informações relativas ao projeto de software, e de um conjunto de serviços e ferramentas, que apóiam as atividades técnicas e gerenciais a serem executadas.

Segundo HARRISON *et al.* [117], pequenos conjuntos de ferramentas isoladas, tais como editores, compiladores e depuradores, constituíram os primeiros esforços para auxiliar desenvolvedores de software na execução de suas atividades. A preocupação com a integração das ferramentas foi o passo seguinte, cujos primeiros resultados foram a melhoria da usabilidade dos conjuntos de ferramentas ainda fracamente integradas e, mais tarde, o surgimento dos Ambientes de Apoio à Programação, com ferramentas fortemente integradas e algum apoio à automação de tarefas. A integração de controle permitia que as atividades de uma ferramenta fossem apropriadamente refletidas em outras ferramentas. Apesar de extremamente úteis, tais

ambientes apoiavam apenas a atividade de implementação. Existiam algumas ferramentas voltadas para produção de artefatos diferentes de código, mas sem integração com outras ferramentas. A identificação da necessidade de suporte integrado para as atividades executadas ao longo do ciclo de vida do software deu origem aos ADS. Novas ferramentas foram desenvolvidas e o tema de pesquisa dominante voltou a ser a integração de ferramentas, desta vez voltada para a integração de dados, uma vez que se tornou necessário o compartilhamento de dados que iam além de textos escritos na linguagem de programação. A definição de uma estrutura que apoiasse a integração, portabilidade e interoperabilidade de ferramentas a partir da integração de dados caracterizou os ambientes dos anos 80 e início dos anos 90 [118]. Para HARRISON *et al.* [117], duas linhas de pesquisa de grande importância na atualidade surgiram na década de 90: os Ambientes de Software Multi-perspectiva, que facilitam a definição e integração/coordenação de diferentes visões simultâneas dos artefatos de software, e os ADS Centrados em Processo, que integram apoio ao desenvolvimento de artefatos de software com apoio à modelagem e à execução dos processos que produzem estes artefatos. ADS Centrados em Processo e ferramentas para apoiar as diferentes atividades da Engenharia de Software foram o foco dos esforços nessa década. SLISKI *et al.* [119] relatam que guiar a integração de ferramentas foi um dos objetivos chave de tais ambientes, propostos por OSTERWEIL [120].

Segundo HARRISON *et al.* [117], o tema da integração talvez seja o mais onipresente nas pesquisas realizadas em ADS. Além da integração de controle e de dados, já mencionadas, a integração de apresentação e de conhecimento também têm sido discutidas. A integração de apresentação [121,122] tem como objetivo possibilitar a interação consistente dos desenvolvedores com o conjunto de ferramentas. A integração de conhecimento [23] busca a integração de ferramentas através do armazenamento, gerenciamento e utilização do conhecimento sobre os métodos, processos e domínios de aplicação manipulados pelas ferramentas.

Além da modelagem de processo e da integração de ferramentas, as pesquisas em ADS têm explorado diferentes aspectos, tais como: assistentes inteligentes, gerência de configuração e versões, desenvolvimento orientado a objetos, reutilização, cooperação e colaboração, arquiteturas de software. OLIVEIRA [20] lista alguns projetos que exploraram estes aspectos. FALBO [123] explora o apoio, baseado em conhecimento, ao processo de software, fornecendo exemplos de uso de assistentes inteligentes e de ADS com suporte baseado em conhecimento. KOBIALKA [124] e ARBAOUI *et al.* [125] apresentam uma revisão comparativa de, respectivamente, dez e oito diferentes ADS Centrados em Processo.

Por fim, pesquisas iniciadas na década de 90 enfatizam a importância de disponibilizar, em ADS, conhecimento relacionado ao domínio de aplicação, com o objetivo de promover o entendimento do domínio e a reutilização e/ou a automação de especificações. Exemplos de tais abordagens podem ser encontrados em [20,126]. Neste contexto, OLIVEIRA [20] definiu os Ambientes de Desenvolvimento de Software Orientados a Domínio (ADSOD) como uma extensão dos ADS Centrados em Processo, na qual projetos de software em domínios específicos são apoiados através do uso do conhecimento do domínio para auxiliar o desenvolvedor no entendimento do problema.

HARRISON *et al.* [117] e FUGGETTA [127] ressaltam, no entanto, que a transferência de tecnologias do contexto de pesquisa para a prática industrial tem sido baixa, enquanto ARBAOUI *et al.* [125] relatam que o número de ADS Centrados em Processo no mercado tem aumentado, apesar de ainda não satisfazerem completamente as demandas do mercado.

Para a década atual, HARRISON *et al.* [117] acreditam na continuidade dos esforços de pesquisa em ADS Centrados em Processo e em ferramentas diversas, particularmente na medida em que aumenta a variedade de domínios desafiadores (tais como Comércio Eletrônico) e modelos de Engenharia de Software (tradicionais e não tradicionais) a serem contemplados. No entanto, destacam a importância de que pesquisas nessas áreas ocorram junto com melhorias na capacidade de adaptação e integração, pois acreditam que o principal desafio para a comunidade é encontrar maneiras de construir e integrar ferramentas de maneira que elas, ou suas funcionalidades, possam facilmente ser adaptadas para uso em novos contextos.

FUGGETTA [127] acredita que a atitude de modelar todos os detalhes de um processo tende a tornar inflexíveis os ADS Centrados em Processo.

Segundo ARBAOUI *et al.* [125], a demanda do mercado hoje pode ser resumida pela necessidade de apoiar processos amplamente distribuídos, heterogêneos, flexíveis e evolutivos.

As idéias de HARRISON *et al.* [117], FUGGETTA [127] e ARBAOUI *et al.* [125] convergem, estabelecendo uma direção para futuros esforços de pesquisa em ADS.

4.3 ADS Centrados em Processo

ADS Centrados em Processo exploram uma representação explícita do processo de software para guiar e auxiliar os desenvolvedores na modelagem e automação de sistemas de software [125]. GARG e JAZAYERI [128] propõem um conjunto de características para os ADS Centrados em Processo:

- *Definição do Processo*: Um engenheiro de software usa o ADS para definir um processo a ser adotado por um ou mais projetos;
- *Análise do Processo*: Um modelo de processo pode ser analisado quanto a sua consistência, completude e correção;
- *Simulação do Processo*: Alguns ADS apóiam a simulação de processo para avaliar a adequação de um processo antes da alocação de recursos para a sua execução;
- *Apresentação do Processo*: O ADS apóia a exibição gráfica do processo, incluindo a informação sobre os produtos gerados, sendo defendido o fornecimento de múltiplas visões do processo, como, por exemplo, uma visão para o gerente e uma para os desenvolvedores;
- *Automação do Processo*: Uma vez que um processo tenha sido definido em um ADS, atividades que não requerem muita intervenção humana podem ser identificadas pelo engenheiro de processo e, subseqüentemente, automatizadas;
- *Orientação sobre o Processo*: O ADS freqüentemente orienta os engenheiros de software, fornecendo-lhe as próximas atividades que podem ser executadas, de acordo com o processo modelado e o estado corrente do processo;
- *Monitoração do Processo*: Um importante aspecto é a capacidade de monitorar a execução de um processo e registrar a história das atividades executadas, de forma a poder utilizá-la em projetos futuros e em esforços para melhoria do processo;
- *Modificação do Processo*: Uma organização, utilizando o ADS, deve ser capaz de modificar suas definições de processo sem interromper o trabalho de suas equipes de projeto;
- *Abertura*: O ADS deve ser capaz de operar em ambientes computacionais nos quais tenha ou não controle sobre todas as ferramentas de software que estão disponíveis;
- *Suporte Multi-usuário*: O ADS tipicamente apóia uma equipe de pessoas trabalhando em um mesmo projeto, assegurando acesso concorrente e sincronizado ao processo e aos dados do produto;
- *Interface do Usuário Específica para a Tarefa*: Dependendo da tarefa a ser executada, o ADS pode limitar as ferramentas que são aplicáveis à tarefa e os produtos visíveis ao usuário.

Segundo KOBIALKA [124], nem todos os ADS Centrados em Processo possuem o mesmo conjunto de características, além das abordagens utilizadas para contemplá-las poderem ser bastante diferentes.

ARBAOUI *et al.* [125] fornecem algumas definições importantes no contexto dos ADS Centrados em Processo, agrupadas segundo os três domínios de processo de software definidos por DOWSON [129]:

- *Definição do Modelo de Processo*, que contém especificações de processo ou de fragmentos de processo, expressos em alguma notação, descrevendo como os processos podem ou devem ser executados. O principal termo é *Modelo de Processo de Software*, que é uma abstração de um processo de software descrita com uma linguagem formal ou semi-formal chamada *Linguagem* ou *Formalismo de Modelagem de Processo*. O *Modelo de Meta-Processo* representa o conjunto de (meta-)atividades para modelar, analisar processos de software e apoiar a evolução dos mesmos;
- *Execução do Modelo de Processo (model enactment)*, que abrange o que ocorre no ADS para apoiar a execução do processo de acordo com o modelo de processo. Significa a execução manual ou automática de um modelo instanciado, o que é atribuição de uma máquina de processo. Um *modelo de processo especializado é instanciado* quando é associado aos produtos e recursos de um projeto concreto, sendo que um *modelo de processo especializado* é o resultado do refinamento e adaptação de um modelo de processo genérico para um projeto específico.
- *Execução do Processo*, que abrange as tarefas e atividades reais que são executadas pelos agentes (humanos ou não) no decorrer de um processo de software. Um *processo de software* é definido pelo conjunto de atividades gerenciais e técnicas executadas na produção e manutenção de software. É um conjunto de atividades parcialmente ordenado, cada uma delas associada com os respectivos artefatos, recursos humanos e computadorizados, restrições e políticas.

Os autores destacam, ainda, a necessidade de apoio à evolução do processo para que o estado de execução do modelo esteja consistente com o estado de execução do processo, uma vez que uma distância significativa entre esses estados significa que o modelo de processo não é mais capaz de influenciar a execução do processo, acarretando a rejeição do ADS Centrado em Processo pelos usuários [130]. No entanto, também destacam que é necessária a tolerância a desvios, uma vez que situações não previstas ocorrem. Neste caso, o ADS tem uma visão parcial do processo real, dependendo completamente do *feedback* fornecido pelas pessoas que estão executando o processo para tornar-se consciente do desvio.

Além disso, ARBAOUI *et al.* [125] propõem o seguinte conjunto de requisitos para ADS Centrados em Processo, considerando os recentes avanços e as atuais necessidades:

- o ADS deve fornecer uma linguagem de modelagem de processo com sintaxe e semântica definidas, bem como suporte à execução do modelo de processo;
- o ADS deve apoiar a ordenação dinâmica das atividades do processo de software;
- o ADS deve apoiar processos de software distribuídos, o que implica em apoiar a interação interpessoal formal e a interoperabilidade entre ferramentas;
- o ADS deve apoiar a evolução do processo de software.

Para KOBIALKA [124], os requisitos são:

- a modificação de um modelo de processo deve ser uma operação simples e segura, porque é impossível definir um processo de software completo desde o início;
- o ambiente deve capacitar o usuário a trabalhar com suporte incompleto ao processo, tanto devido a conhecimento incompleto sobre o processo quanto pelo desejo de não especificá-lo em muitos detalhes;
- o usuário deve ser capaz de complementar ou compensar os efeitos da execução de um modelo de processo através da execução de qualquer ação que possa estar em um modelo de processo;
- o usuário deve ser capaz de facilmente entender os efeitos da execução de um modelo de processo, o que significa que a linguagem de modelagem de processo deve ser muito simples de usar e entender;
- o impacto de um processo tem que poder ser analisado com antecedência;
- o suporte a processo deve ser adaptável para (sub-)projetos específicos;
- deve haver meios de definir e controlar permissões para modificação dos processos.

KOBIALKA [124] acredita, ainda, que a comunicação e o fluxo de trabalho devem ser apoiados em primeiro lugar, de preferência em termos de orientação (*guidance*) e somente em termos de algo a ser obrigatoriamente cumprido (*enforcement*) nos casos em que o processo é completamente entendido; decisões humanas são de grande importância em quase todos os passos do processo de software, pois fatores técnicos e não técnicos podem influenciar o processo de forma não-determinística; visibilidade e rastreabilidade são essenciais tanto para as pessoas quanto para a execução dos modelos de processo.

No que se refere às linguagens de modelagem de processo, também chamadas de linguagens de definição de processo ou linguagens de descrição de processo [119,124,125,127,131,132,133,134]:

- existem diferentes tipos [127] e muitos ADS Centrados em Processo fornecem uma combinação de abordagens para descrição dos processos [124];

- podem ser usadas com o objetivo de definição do processo, entendimento do processo, treinamento e educação, simulação e otimização e para fornecer suporte ao processo [127].

KOBIALKA [124], SLISKI *et al.* [119], MAURER e HOLZ [135] citam tipos de linguagens utilizadas para modelagem de processos de software: procedimentais, baseadas em regras, funcionais hierárquicas, baseadas em diagramas de fluxo de dados, baseadas em rede Petri, orientadas a objetos, além de variações dos tipos de linguagens já citados. KOBIALKA [124], SLISKI *et al.* [119] e ARBAOUI *et al.* [125] fornecem exemplos de ADS Centrados em Processo que utilizam os tipos de linguagens mencionados.

Segundo FUGGETTA [127], uma série de linguagens e formalismos de modelagem de processo têm sido criados, mas as linguagens existentes são complexas, extremamente sofisticadas e fortemente orientadas à modelagem detalhada dos processos, pois visam a execução e, freqüentemente, a automação dos processos. Estas características limitam as possibilidades de adoção das linguagens na prática, onde o mais importante é a comunicação e entendimento dos processos. Para o autor, linguagens de modelagem de processo devem ser fáceis de usar, intuitivas e tolerantes, de modo que seja possível construir modelos de processos de forma incremental.

Por fim, FUGGETTA [127] recomenda que os ADS Centrados em Processo:

- não sejam intrusivos, ou seja, suavemente se integrem e complementem o ambiente de desenvolvimento tradicional, automatizando de maneira efetiva apenas os fragmentos de processo que são razoáveis automatizar;
- sejam capazes de tolerar e gerenciar inconsistências e desvios, de forma a refletir a natureza criativa da atividade de desenvolvimento de software;
- informem claramente os desenvolvedores sobre o estado do processo de desenvolvimento de software de muitos pontos de vista diferentes;
- sejam implantados de forma incremental, de modo que a transição para a nova tecnologia seja facilitada e riscos sejam reduzidos.

4.4 ADS Orientados a Domínio

Um Ambiente de Desenvolvimento de Software Orientado a Domínio (ADSOD) [20,136] fornece conhecimento sobre um domínio específico como elemento central da assistência aos desenvolvedores durante o processo de software, onde um domínio é uma área de aplicação (como, por exemplo, cardiologia) para a qual vários produtos de software podem ser desenvolvidos. A concepção de tais ambientes baseou-se na

premissa de que uma das principais dificuldades no desenvolvimento de sistemas é o fato dos desenvolvedores de software não conhecerem ou não estarem familiarizados com o domínio para o qual devem desenvolver um determinado sistema. Desta forma, um ADSOD, além do repositório e do conjunto de serviços e ferramentas, possui duas outras características essenciais: a incorporação de conhecimento sobre o domínio e a viabilização do uso deste conhecimento ao longo do processo de software.

A figura 4.1 ilustra os elementos básicos de um ADSOD, que são o reflexo de três questões: (i) qual conhecimento deve estar disponível no ambiente, (ii) como este conhecimento deve estar organizado e representado, (iii) quando e como utilizar este conhecimento durante o desenvolvimento e a manutenção de software.

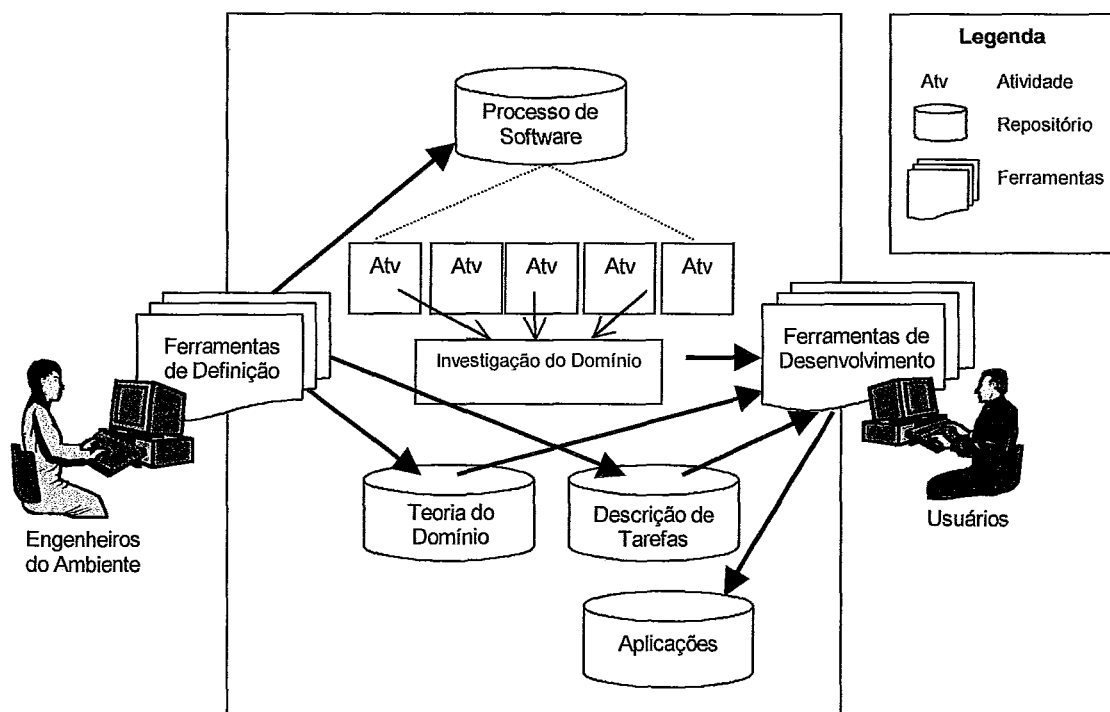


Figura 4.1 – Visão Geral de um ADSOD [20]

A *Teoria do Domínio* e a *Descrição de Tarefas* foram concebidos para responder as questões (i) e (ii). A *Teoria do Domínio* contém o conhecimento do domínio a ser utilizado no ambiente, que referencia descrições de tarefas relacionadas aos conceitos para facilitar o entendimento do domínio. Para responder a questão (iii), deve-se considerar o estudo e investigação do domínio nas diferentes atividades do *Processo de Software*, de forma a apoiar a execução das mesmas. Estes diferentes elementos são incorporados ao ADSOD através de *Ferramentas de Definição* e são utilizados ao longo de um projeto específico através das *Ferramentas de Desenvolvimento*. Os engenheiros do ambiente definem o processo de software e a *Teoria do Domínio* para o ADSOD a ser construído. Além disso, definem tarefas que podem ser aplicadas àquele e a outros domínios. Uma vez concluído o ambiente, os usuários de

desenvolvimento, através de ferramentas específicas, acessam e utilizam a Teoria do Domínio em cada uma das atividades do processo de software.

A *Teoria do Domínio* é composta basicamente da organização de conceitos, propriedades e restrições (axiomas) do domínio através de ontologias do domínio e do mapeamento dos conceitos com tarefas pertinentes no domínio considerado. Uma Teoria do Domínio é dividida em sub-teorias e uma sub-teoria contém a ontologia sobre a parte específica do domínio à qual se refere a sub-teoria. À medida que diferentes aplicações vão sendo desenvolvidas no ADSOD, a Teoria do Domínio pode evoluir, incorporando mais conhecimento sobre o domínio, quer seja através da instanciação dos conceitos ou da inclusão de novos conceitos e respectivas relações e restrições.

Uma tarefa é, geralmente, independente do domínio, sendo genérica e aplicável a diferentes domínios. Para a construção de um ADSOD, pressupõe-se a existência e contínua definição de um repositório de descrições de tarefas genéricas. Na *Descrição de Tarefas* proposta por OLIVEIRA [20], cada tarefa é descrita em termos de uma descrição de alto nível da tarefa; uma ontologia dos conceitos da tarefa e um conjunto de referências bibliográficas. ZLOT [137], posteriormente, refinou esta estrutura, propondo que a ontologia da tarefa seja responsável por descrever os conceitos e relações envolvidos na solução da tarefa e que a solução da tarefa seja fornecida através de sua decomposição em sub-tarefas ou pela descrição da inferência utilizada para resolvê-la. As sub-tarefas são descritas da mesma forma que as tarefas que as originaram. ZLOT [137] também propõe que a descrição da tarefa seja organizada em três níveis de abstração (Verbal, Conceitual e Formal).

Com relação ao *Processo de Software*, OLIVEIRA [20] cita as principais atividades do desenvolvimento de software, destacando que essas atividades dependem direta ou indiretamente do conhecimento do domínio, sendo que algumas atividades são mais orientadas a domínio do que outras. Desta forma, definiu a sub-atividade *Investigação do Domínio*, que pode ser incluída em qualquer atividade do processo de software em que seja importante o entendimento e uso do conhecimento do domínio para auxiliar a execução da atividade.

Segundo OLIVEIRA [20], tanto a introdução do conhecimento do domínio quanto o seu uso em um ADSOD deve ser apoiado por ferramentas e, além disso, devem ser fornecidas ferramentas específicas para um domínio, ou seja, ferramentas que utilizam a Teoria do Domínio como linguagem de interação com o usuário. Desta forma, as *Ferramentas de Definição* (aquelas que permitem a construção dos ADSOD) consideradas importantes pela autora foram: um editor de Teorias de Domínio, um editor de tarefas e um assistente para definição de processos de software. Em termos

de *Ferramentas de Desenvolvimento* relevantes em um ADSOD, tem-se: um assistente para aprendizado do domínio, que permita a pesquisa de conceitos do domínio através da estrutura hierárquica e das relações entre os mesmos; uma ferramenta para elicitaco especfica do domnio, ou seja, uma ferramenta baseada no uso da terminologia definida pela Teoria do Domnio para interao com os especialistas do domnio; uma ferramenta de apoio à modelagem que auxilie o desenvolvedor a reutilizar as definies e a estrutura de organizao dos conceitos estabelecidas na Teoria do Domnio; uma ferramenta para registro do uso dos conceitos em um projeto, incluindo exemplos de instncias dos conceitos utilizados; e uma ferramenta para evoluo da Teoria do Domnio, que apie o registro de novos conceitos na Teoria do Domnio e das respectivas relaes e restries.

4.5 Projeto TABA

O Projeto TABA foi criado com o objetivo de desenvolver uma estao de trabalho capaz de auxiliar na definio, construo e execuo de ADS adequados às particularidades de domnios de aplicao e projetos de software especficos [22,23]. Sendo assim, a Estao TABA possua as seguintes funes:

- (a) auxiliar o engenheiro de software na especificao e instanciao do ambiente mais adequado ao desenvolvimento de um produto especfico;
- (b) auxiliar o engenheiro de software a implementar as ferramentas necessrias ao ambiente definido em (a);
- (c) permitir o desenvolvimento do produto de software atravs do ambiente instanciado em (a);
- (d) permitir a execuo do produto de software na prpria estao.

Para contemplar estas funes, foram concebidos quatro ambientes [22,23]:

- ambiente especificador e instanciador de ADS (Meta-ambiente TABA), responsvel pela funo (a);
- ambiente de construo de ferramentas, responsvel pela funo (b) e pela incorporao das ferramentas no Meta-ambiente TABA;
- ambiente de desenvolvimento (ADS Instanciado), definido e instanciado atravs do Meta-ambiente, sendo responsvel pela funo (c);
- ambiente de execuo, responsvel pela funo (d).

4.5.1 Estrutura e Filosofia de Integração

A estrutura da Estação TABA [23] foi definida como um conjunto de componentes integrados que possuem controle sobre a sua existência, suas informações, estados e funcionalidades básicas, os quais são:

- *Sistema Computacional*, representado pelo hardware e pelo sistema operacional;
- componente *Interface com o Usuário*, responsável pelo controle e gerenciamento da interação do usuário com o ADS;
- componente *Cooperação*, que trata das necessidades de comunicação e interação entres os membros da equipe de projeto e estende o componente de Interface com o Usuário para tornar possível a interface de grupo;
- componente *Controle de Processo*, responsável pelo controle e gerenciamento do processo de desenvolvimento através da identificação de atividades, gerência da execução das ferramentas e controle dos papéis e atividades dos usuários;
- componente *Suporte Inteligente*, que fornece inteligência global ao ambiente, apoiando o usuário na utilização do Meta-ambiente e dos ADS Instanciados;
- componente *Reutilização*, que fornece mecanismos que possibilitam a reutilização de especificações anteriores de ADS, de componentes para a construção de ferramentas, bem como de componentes para a construção de um produto de software;
- componente *Conhecimento*, que incorpora mecanismos para o armazenamento e a utilização de conhecimento, que podem ser utilizados tanto pelo Meta-ambiente quanto pelos ADS Instanciados, sendo responsável pela integração de conhecimento;
- componente *Repositório Comum*, responsável pelo controle, gerenciamento e armazenamento dos objetos manuseados por cada ADS Instanciado e pelo Meta-ambiente, e pela consistência e integridade das informações;
- componente *Controle de Versões*, que controla e gerencia versões de documentos e itens de software produzidos.

Esses componentes definem a filosofia de integração tanto da Estação TABA quanto dos seus ambientes instanciados, o que permite a integração de ferramentas internas e fornece recursos para a incorporação de ferramentas externas. A filosofia de integração envolve quatro dimensões [23]:

- integração de apresentação e interação, que é responsabilidade do componente *Interface com o Usuário*, possibilitando a homogeneização das formas de apresentação das informações e das técnicas de interação com o usuário;

- integração de controle e gerenciamento, que é responsabilidade dos componentes *Cooperação*, *Controle de Processos*, *Reutilização* e *Controle de Versões*, fornecendo serviços e funcionalidades básicas que permitem o funcionamento, de forma organizada, dos ADS Instanciados e de suas ferramentas;
- integração de dados, que é responsabilidade do componente *Repositório Comum*, fornecendo serviços básicos de armazenamento e gerenciamento de estruturas de informação obtidas a partir das ferramentas que compõem a Estação,
- integração de conhecimento, que torna possível os serviços básicos de armazenamento, gerenciamento e utilização do conhecimento descrito na Estação TABA ou adquirido ao longo do processo de software.

A integração de conhecimento na Estação TABA foi posteriormente tratada por FALBO [123], que propôs o uso de servidores de conhecimento, nos quais o conhecimento é modelado para reutilização e disponibilizado em um conjunto de componentes que podem ser utilizados por várias ferramentas a serem desenvolvidas. No que se refere ao conhecimento sobre tarefas, um Servidor de Conhecimento deve fornecer uma biblioteca de resolvedores genéricos de problemas, passíveis de serem instanciados e adaptados para aplicações particulares. No que se refere ao conhecimento sobre domínios de interesse, é desejável que a base de conhecimento do Servidor seja modular e baseada em ontologias. Desta forma, a arquitetura de um Servidor de Conhecimento é composta de:

- uma base de conhecimento modular, onde cada módulo de conhecimento contém um corpo de conhecimento reutilizável, construído com base em uma ontologia;
- a máquina de inferência do sistema de representação adotado, associada a um conjunto de *templates* de resolvedores de problemas capazes de especializá-la para os tipos de problema mais freqüentemente encontrados no universo de discurso.

A base de conhecimento modular é essencialmente uma biblioteca de ontologias, pois cada ontologia é implementada em um módulo de conhecimento, sendo que suas instanciações também dão origem a módulos de conhecimento. Assim, a base de conhecimento de uma aplicação que se comprometa com uma ou várias ontologias pode vir a ser a conjunção dos módulos de conhecimento correspondentes, mais o conhecimento específico da aplicação particular. Além disso, os módulos de conhecimento e os *templates* de resolvedores de problema estão estreitamente relacionados: os papéis considerados em um *template* de resolvidor de problema são preenchidos com o conhecimento dos domínios descritos nos módulos de conhecimento. Por exemplo, sensibilidade, um conceito do domínio de acústica

submarina, desempenha o papel de parâmetro para o problema de configuração de um sonar [137].

No que se refere à tecnologia de representação utilizada na construção de Servidores de Conhecimento, a Estação TABA fornece uma classe de representação de conhecimento que permite a definição de diferentes tecnologias de representação: rede neural e sistemas lógicos, englobando *frames*; sistema de programação em lógica e rede semântica. Atualmente, está implementada a classe de sistema de programação em lógica utilizando Prolog [123].

4.5.2 Definição e Execução de Processo

A instanciação de um ADS na Estação TABA é feita a partir da definição de um processo de desenvolvimento que se caracteriza pela descrição de uma seqüência de atividades, suas ferramentas de apoio, produtos de software gerados e recursos consumidos. OLIVEIRA [20] apresenta um resumo de importantes trabalhos que foram realizados no sentido de apoiar não só a definição do processo de desenvolvimento, como também o acompanhamento e alteração do processo durante a sua execução. VASCONCELOS [138] propõe a utilização de padrões para organizar as descrições de processos, possibilitando a alteração e reutilização dos processos definidos. ARAÚJO [139] propõe uma máquina de processo e estabelece uma estrutura que permite adaptações do processo ao longo de sua execução. As alterações são válidas a partir do ponto corrente de execução do processo. FALBO [123] acrescenta assistência inteligente na definição do processo, construindo um Servidor de Conhecimento de Processo, com uma ontologia sobre processo de software embutida, e uma ferramenta, ASSIST-PRO, que será descrita posteriormente.

OLIVEIRA [20] propõe a utilização de Processo Padrão¹ para a definição de processos na Estação TABA, onde o objetivo é estabelecer um processo de desenvolvimento comum a ser utilizado por uma organização, a partir do qual processos de desenvolvimento de software específicos possam ser definidos e, conseqüentemente, os ADS específicos para os projetos. OLIVEIRA [20] destaca como premissas:

- A norma ISO/IEC 12207 [140], que estabelece os processos do ciclo de vida de um software, foi definida para um projeto genérico e deve ser adaptada para

¹ Processo Padrão, segundo o SPICE (*Software Process Improvement and Capability dEtermination*) [141], é a definição operacional do processo básico que guia o estabelecimento de um processo comum dentro da organização. O Processo Padrão descreve os elementos fundamentais que devem ser incorporados em qualquer processo definido e as relações entre esses elementos (como seqüência e interfaces).

organizações e projetos específicos. Para a adaptação, é necessário identificar as características do ambiente do projeto;

- É preciso considerar a cultura da organização, o modo como ela desenvolve software e os problemas cruciais que são enfrentados em seus projetos de desenvolvimento de software, contemplando-os no processo;
- Como diferentes tipos de software (por exemplo: sistemas de informação, sistemas baseados em conhecimento, aplicações multimídia) podem ser desenvolvidos para um mesmo domínio e em uma mesma organização, torna-se, então, necessário definir diversos processos de desenvolvimento de software adequados para a construção dos diferentes tipos de software;
- Características do projeto, como requisitos, tamanho, quantidade de pessoas e partes envolvidas, também devem ser consideradas em um processo de desenvolvimento de software para um projeto específico;
- Definir vários processos de software para uma mesma organização pode ser muito trabalhoso e ter como resultado processos completamente distintos, o que não é adequado. Pesquisas mostraram a necessidade de padronização dos processos de software dentro de uma organização [140,141,142,143].

Dessa forma, OLIVEIRA [20] conclui que um processo de desenvolvimento de software, a ser definido na Estação TABA para instanciar um ADS, deve considerar a definição de um Processo Padrão para a organização para a qual o ambiente será construído, que, por sua vez, deve servir de base para a definição dos processos de software adequados aos diferentes tipos de software desenvolvidos na organização e, a partir dos quais, processos para projetos específicos podem ser definidos. O modelo para definição de processos de software na Estação TABA é detalhado em [144].

Segundo a proposta do modelo, que estabelece etapas e produtos intermediários, a norma ISO/IEC 12207 [140] é a base para a definição de qualquer Processo Padrão a partir da Estação TABA, podendo, também, ser considerado um dos modelos de maturidade: CMM (*Capability Maturity Model*) [145] ou SPICE [141]. Para a definição do Processo Padrão, são, ainda, consideradas as características do desenvolvimento de software na organização, que são relacionadas ao ambiente de trabalho, conhecimento e experiência das equipes envolvidas e à própria cultura e experiência da organização no desenvolvimento de software. A partir do Processo Padrão, diferentes processos de software podem ser especializados de acordo com as características dos tipos de software produzidos na organização (por exemplo: sistemas especialistas, sistemas de informação e sistemas de controle de processos) e dos paradigmas de desenvolvimento adotados (por exemplo: orientado objeto e

estruturado). Neste momento, novas atividades podem ser definidas e incluídas nos processos especializados e a descrição de atividades já definidas no Processo Padrão pode ser adequada. No entanto, todos os elementos básicos definidos no Processo Padrão deverão sempre estar presentes nos processos especializados. Para ser utilizado em um projeto, o processo especializado mais adequado para um determinado tipo de software deve ser instanciado para atender às características do projeto específico, devendo-se considerar o tamanho e a complexidade do produto bem como as características de qualidade desejadas, a expectativa de vida útil, as características da equipe de desenvolvimento e demais características do projeto. Neste momento, são selecionados o modelo de ciclo de vida, os métodos e ferramentas. O processo de software instanciado final é, então, utilizado para instanciar o ADS na Estação.

4.5.3 Instanciação de ADSOD

OLIVEIRA [20] estendeu a Estação TABA para possibilitar a especificação e instanciação de ADSOD para qualquer domínio desejado. Na verdade, a Estação TABA, inicialmente implementada em Eiffel para a plataforma Unix da *Sun Workstation*, foi re-implementada em C++ para plataforma Windows. A re-implementação não contemplou todo o trabalho realizado desde a criação do projeto TABA, o que seria inviável em curto espaço de tempo, contemplando as funcionalidades necessárias à especificação, instanciação e execução dos ambientes. Desta forma, a Estação TABA tornou-se capaz de instanciar ADS a partir do processo de software ou a partir do processo de software e do domínio de aplicação.

As principais facilidades fornecidas pela Estação TABA passaram a ser [20]: apoio à descrição do conhecimento de processo, apoio à definição de processos de software, apoio à descrição de tarefas, apoio à descrição da Teoria do Domínio, gerência de ferramentas e instanciação de ADS convencionais ou ADSOD. *Apoio à descrição de conhecimento de processo* significa apoiar a introdução de descrições gerais de atividades, processos, métodos, técnicas e demais aspectos necessários para auxiliar na definição de um processo de software. *Apoio à definição de processos de software* refere-se à definição propriamente dita dos processos necessários à instanciação dos ambientes. *Apoio à descrição de tarefas* implica em apoiar a descrição de tarefas a serem utilizadas nos ADSOD. O *apoio à definição da Teoria do Domínio* implica em permitir a introdução da Teoria do Domínio específica para que seja possível a instanciação de um ADSOD. Este apoio inclui a geração dos módulos de conhecimento para o domínio específico, dando origem a um Servidor de Conhecimento para o Domínio. A *Gerência de ferramentas* refere-se às facilidades da

Estação TABA para integrar ferramentas conforme definido por TRAVASSOS [23], permitindo o seu uso na Estação e em seus ambientes instanciados. A *Instanciação de ADS* refere-se à geração do ambiente instanciado, orientado a domínio ou não, de acordo com as características do processo previamente definidas.

A funcionalidade básica dos ADS Instanciados é *gerenciar o desenvolvimento do software específico*, ou seja, apoiar o acompanhamento, controle e alteração do processo de software durante o desenvolvimento, no que se refere à inclusão de novas atividades e sub-atividades, conforme definido por ARAUJO [139]. Nos ADSOD, além da gerência do desenvolvimento, estão presentes também as funcionalidades *orientação ao desenvolvimento no domínio*, considerando as características do domínio e utilizando ferramentas específicas de investigação do domínio, e *gerência dos componentes do domínio*. Esta função, não-automatizada, consiste da avaliação feita pelo engenheiro do domínio sobre o conhecimento e os componentes gerados em cada projeto com fins de evolução da Teoria do Domínio, verificação da necessidade de descrição de novas tarefas e identificação dos componentes reutilizáveis [20].

As ferramentas que implementam as funcionalidades acima podem ser agrupadas em ferramentas para definição de ambientes e ferramentas de apoio ao processo de software. Dentre as ferramentas para definição de ambientes, tem-se:

- um editor de tarefas (EDITAR) [137], que permite a descrição de tarefas genéricas, independentes de domínio, mas que são importantes para o entendimento de como e para que os conceitos do domínio são utilizados. Esta ferramenta contempla: (i) uma descrição de alto nível da tarefa; (ii) a especificação das sub-tarefas que a compõem, caso a tarefa seja complexa; (iii) a definição da ontologia da tarefa, contendo os conceitos requeridos ou produzidos pela solução do problema representado pela tarefa, junto com as relações e restrições pertinentes; (iv) as inferências necessárias para que a tarefa possa ser executada; e (v) um conjunto de referências bibliográficas. As restrições não geradas automaticamente pela ferramenta e as inferências são fornecidas pelo usuário de maneira informal e em Prolog.
- um editor de Teoria do Domínio (EDITED) [20], que é utilizado para permitir a introdução do conhecimento do domínio na Estação TABA, de forma que todos os conceitos e relações do domínio sejam organizados de acordo com a estrutura definida pelos servidores de conhecimento, disponibilizando módulos de conhecimento do domínio passíveis de serem utilizados nos ambientes instanciados. A ferramenta (i) permite a entrada de informações sobre o domínio, como conceitos, relações entre os conceitos, propriedades dos conceitos, eventos

que modificam os valores de propriedades, entre outros; (ii) permite a entrada de restrições do domínio de maneira informal e descritas em Prolog (conhecimento ontológico), (iii) gera o conhecimento ontológico em um Servidor de Conhecimento do Domínio específico para o domínio em questão e (iv) gera as classes para cada conceito do domínio, nas quais serão registradas as instâncias desses conceitos.

- um editor de processo (EDIT-PRO) [20], que é uma ferramenta para simples introdução, na Estação TABA, de um processo previamente definido pelo engenheiro de software. O usuário inclui as características do projeto e descreve o processo definido para o projeto. Para isto, escolhe, do conhecimento sobre processo de desenvolvimento de software disponível na Estação, o modelo de ciclo de vida, as atividades, os métodos, as ferramentas, os recursos e os produtos, e estabelece a seqüência para execução das atividades. EDIT-PRO permite, ainda, customizar os nomes das atividades e de recursos para o processo específico que está sendo definido, tornando possível, assim, a utilização de nomes já comumente utilizados na organização;
- um assistente inteligente para definição de processo (ASSIST-PRO) [123], que auxilia o engenheiro de software na descrição do processo específico para o ambiente a ser instanciado. ASSIST-PRO entrevista o engenheiro de software, procurando identificar características do projeto a ser desenvolvido, e sugere ciclos de vida e atividades que mais se adequam ao problema;
- uma ferramenta para definição de processo (DEF-PRO) [144], com o objetivo de apoiar desde a definição do processo padrão até a definição do processo instanciado, conforme o modelo proposto por OLIVEIRA [20], considerando não apenas a norma ISO/IEC 12207 [140], modelos de maturidade CMM [145] e SPICE [141], mas, também, as características da organização, tipo de software, paradigma de desenvolvimento e o tipo de ambiente que se deseja instanciar (orientado ao domínio ou não). A ferramenta apóia o engenheiro do ambiente na definição dos processos, fornecendo-lhe o conhecimento registrado na Estação sobre as atividades associadas aos elementos acima mencionados (normas, modelos, tipos de software, etc).

Já em termos de ferramentas de apoio ao processo de software, tem-se:

- uma ferramenta para identificação de requisitos de qualidade para produtos de software (Q-Fuzzy) [146], que pode ser customizada para cada ADSOD instanciado na Estação TABA [20]. A ferramenta baseia-se no modelo *fuzzy* para identificação de requisitos de qualidade proposto por BELCHIOR [147], que define quatro etapas: (i) identificação do objeto a ser avaliado e do conjunto de características de

qualidade a ser considerado, (ii) escolha de especialistas e definição do perfil dos mesmos, (iii) determinação, por cada especialista, do grau de importância de cada característica de qualidade identificada na primeira etapa e, por fim, (iv) tratamento dos dados coletados na avaliação de cada característica de qualidade para gerar um consenso entre os especialistas. O perfil do especialista é utilizado para determinar o peso de suas opiniões e o consenso a partir das diferentes opiniões é obtido a partir de uma função *fuzzy* específica. A customização da ferramenta para um ADSOD específico ocorre através da inclusão de características de qualidade referentes ao tipo de software e ao domínio de aplicação do projeto, além da inclusão de questionários para identificação do perfil dos especialistas, que são específicos da organização;

- uma ferramenta para apoiar o aprendizado sobre o domínio (NAVEGUE) e duas ferramentas auxiliares (GENESIS e REGCON) [148], que, respectivamente, apóiam o registro de instâncias de conceitos da Teoria do Domínio e o registro do uso desses conceitos. O desenvolvedor utiliza a REGCON para identificar quais conceitos da Teoria do Domínio foram utilizados em um determinado projeto. Através da ferramenta GENESIS, o desenvolvedor registra exemplos dos conceitos, que são um elemento importante para o entendimento do conhecimento sobre o domínio, fornecendo, também, os valores das propriedades dos conceitos que caracterizam os exemplos. Com NAVEGUE, o desenvolvedor tem acesso, através de uma estrutura hierárquica, (i) à descrição da Teoria do Domínio; (ii) a sua composição em termos de sub-teorias com respectivas descrições e tarefas relacionadas; (iii) a cada um dos conceitos que compõem as sub-teorias, com sua descrição, relações com outros conceitos, projetos em que foi utilizado e exemplos; e (iv) a cada atributo dos conceitos, com respectiva descrição, unidade e valores. As três ferramentas visam apoiar o desenvolvedor na atividade de Investigação do Domínio.

Além disso, GOMES [149] acrescentou à Estação TABA um conjunto de funcionalidades para permitir a avaliação do processo de *software* ao final do projeto. A avaliação é baseada em métricas, tendo sido: (i) selecionado um conjunto de métricas para avaliar o processo com o objetivo de melhorar a precisão das estimativas de projeto; (ii) incorporada a coleta dos dados ao processo produtivo nos ambientes instanciados; e (iii) definida uma estrutura de decisão capaz de orientar o procedimento de avaliação do processo de acordo com o objetivo mencionado. A coleta de dados é realizada através do preenchimento de documentos pelo gerente do projeto e pelos desenvolvedores ao longo do processo. A análise dos resultados das

medições é feita por um sistema baseado em conhecimento a partir dos valores de tolerância fornecidos pelo gerente do projeto. A estrutura proposta para a avaliação de processos é extensível e novas funcionalidades podem ser acrescentadas para possibilitar a avaliação segundo outros objetivos, como os outros dois objetivos estabelecidos pelo autor: melhorar a qualidade dos produtos liberados para uso e diminuir o custo final dos projetos.

Na versão em Eiffel da Estação TABA, foram instanciados dois ADS: ORIXÁS [150], com o objetivo de apoiar o desenvolvimento de sistemas baseados em conhecimento, e MEMPHIS [151], com o objetivo de apoiar o desenvolvimento de software baseado em reutilização. A versão em C++ foi utilizada para a definição e instanciação de três ADSOD: CORDIS [20], para o domínio de cardiologia, NETUNO [148], para o domínio de Acústica Submarina, e INSECTA [152], para o domínio de entomologia².

4.6 Projeto MILOS

MILOS [94] é um projeto da Universidade de Calgary, no Canadá, com a Universidade de Kaiserslautern, na Alemanha, que tem como objetivo fornecer uma infra-estrutura que integre os conceitos de ADS Centrado em Processo e de Gerência do Conhecimento.

Sendo assim, o sistema é composto de *Ambiente de Modelagem Estendida de Processos*, *Ambiente de Planejamento de Projeto*, *Ambiente de Workflow* e *Assistente de Informação*, que armazenam e manipulam [6,94,135]:

- Modelos Genéricos de Processo, que são descrições reutilizáveis do *workflow* de processos de desenvolvimento de software. Tais modelos contêm associações com itens de conhecimento que foram considerados úteis no passado, que podem ser: modelos de habilidades, conhecimento de contexto ou modelos de necessidade de informação;
- Planos de Projeto, que são adaptações de modelos de processos genéricos de acordo com as necessidades dos projetos específicos. Os elementos dos planos herdam as associações com itens de conhecimento registradas nos modelos de processo;
- Dados de Projeto, que são dados criados durante a execução do processo;
- Modelos de Habilidades, que podem ser anexados às tarefas. O conceito de habilidade é considerado de forma abrangente (uma habilidade pode ser geral,

² Entomologia é uma disciplina da agronomia/zoologia que estuda os insetos.

técnica ou referente ao uso de uma ferramenta) e as habilidades são organizadas em uma ontologia;

- Conhecimentos de Contexto, que podem ser anexados a processos e são qualquer tipo de documento acessível através da Web. A equipe do projeto é capaz de anexar associações ao processo, que passam a ser compartilhadas pelos desenvolvedores que estão executando tarefas do mesmo tipo que a considerada;
- Modelos de Necessidade de Informação, que representam necessidades de informação que podem surgir durante o planejamento ou execução de processos específicos. Para isto, fazem referência a informações que ficam disponíveis somente quando a tarefa está sendo executada, sendo necessário a definição de consultas que podem conter variáveis cujos valores serão fornecidos pela máquina de *workflow*.

Além disso, dois componentes fundamentais do MILOS são [94,135]:

- Máquina de *Workflow*, que dá suporte à execução dos planos de projeto, gerenciando o estado do processo de trabalho e das tarefas envolvidas, as listas de atividades a serem fornecidas para os usuários, os produtos criados durante a execução do processo, entre outros;
- Assistente de Informação, que utiliza os modelos de necessidade de informação para submeter consultas dinâmicas às fontes de informações apropriadas, das quais as respostas podem ser recuperadas.

O uso do MILOS é descrito como um processo de 4 passos básicos [6,94,135]. No primeiro passo, o gerente do projeto cria um plano de projeto inicial selecionando tipos de processo e métodos da biblioteca de modelos de processo. Uma vez estabelecido um método, o gerente de projeto consulta o conjunto de recursos disponíveis, procurando pessoas que tenham experiência com o método. Ele também pode pesquisar por instâncias anteriores do mesmo tipo de processo na base de dados de planos de projeto. O plano do projeto pode ser modificado a qualquer momento durante a execução do processo, porque o planejamento do projeto é incremental. O plano de projeto corrente é a base para a máquina de *workflow* (passo 2). Um usuário é capaz de conectar-se à máquina de *workflow* e ter acesso a seu espaço de trabalho individual, que fornece a lista de tarefas a serem feitas; as descrições das tarefas e as informações associadas; além de informações sobre cronograma, permitindo o usuário aceitar, rejeitar ou reiniciar uma tarefa; acessar as entradas da tarefa e editar as saídas; além de informar o percentual da atividade que já foi concluído. MILOS fornece ao usuário uma estimativa de duração das tarefas de acordo com a velocidade média do usuário na execução dessas tarefas, permitindo a

previsão das datas de término das mesmas. No caso desta previsão violar o cronograma do projeto, o gerente de projeto é notificado do problema. O papel do Assistente de Informação no planejamento e na execução das tarefas é efetuar consultas sob demanda às fontes de informação, distribuindo os resultados para os usuários, além de permitir que eles formulem questões que representam novas necessidades de informação. Estas novas necessidades de informação são associadas com a tarefa no contexto em que tenham ocorrido e o gerente do projeto é notificado. MILOS usa um sistema de controle de versão para gerenciar todos os arquivos que são produzidos e acessados em um projeto. O terceiro passo é extrair conhecimento reutilizável sobre processo a partir do plano de projeto corrente. Um usuário pode selecionar partes do plano do projeto e integrá-las à biblioteca de modelos de processo (passo 4). Durante o empacotamento da experiência, os membros do grupo de processo de software são alertados sobre as necessidades de informação adicionadas ao plano e completam os modelos de necessidade de informação parcialmente definidos.

MAURER e HOLZ [135] informam que existem dois protótipos de implementação do MILOS, sendo que o primeiro implementa apenas o componente de suporte a processos e o segundo implementa os componentes de suporte a processos, modelagem das necessidades de informação e o Assistente de Informação.

4.7 Projeto ODE

Ontology-based software Development Environment (ODE) é um projeto da Universidade Federal do Espírito Santo [87,153]. O ambiente é um ADS Centrado em Processo e baseado em ontologias que possui uma infra-estrutura de gerência do conhecimento.

No ODE, ontologias devem ser construídas para os sub-domínios de Engenharia de *Software* de forma a serem utilizadas para promover a integração de ferramentas e permitir uma visão orientada a conteúdo da gerência do conhecimento. A partir das ontologias, são derivados *frameworks*, que são utilizados como base para a construção e integração de ferramentas. Cada ferramenta ou serviço a ser desenvolvido deve usar e especializar os *frameworks* derivados.

Como o ODE é um ADS Centrado em Processo, uma ontologia de processo de *software* [123] foi utilizada para derivar o *framework* para construção da infra-estrutura de processo e, também, para construção das ferramentas de definição de processo e acompanhamento de projeto. Além disso, todas as ferramentas internas do ambiente compartilham o vocabulário definido por esta ontologia.

Segundo BERTOLLO e FALBO [154], a ferramenta de definição de processo fornece diretrizes para o gerente do projeto, que aceita ou rejeita as sugestões, e permite a definição de processos em níveis, com base no modelo proposto por OLIVEIRA [20]. Para acompanhamento do projeto, ODE fornece uma ferramenta que apresenta o processo e o estado de suas atividades.

Seguindo a mesma abordagem, uma ontologia de qualidade de software foi desenvolvida e utilizada na construção de ferramentas para gerência da qualidade. *ControlQ* é uma ferramenta que apóia o planejamento e a avaliação da qualidade [87,155].

Uma outra ferramenta do ambiente é ODEd, um editor de ontologias que apóia, através de representações gráficas, a definição de conceitos e relações; gera automaticamente algumas classes de axiomas e apóia a derivação de *frameworks* a partir das ontologias e a instanciação das mesmas. Esta ferramenta permite que ontologias de domínio também sejam inseridas no ambiente e prevê a geração automática de hipertextos baseados nas ontologias definidas [156].

A infra-estrutura do ODE para gerência do conhecimento é composta de uma memória organizacional e de serviços de apoio à gerência do conhecimento [87,155]. A memória organizacional contém três tipos de itens de conhecimento: artefatos, instâncias de ontologias e lições aprendidas. Os serviços de gerência do conhecimento são agrupados em serviços gerais, que são diretamente incorporados ao ambiente, e serviços específicos das ferramentas, que são definidos de acordo com as características específicas de cada ferramenta. Os serviços gerais referem-se à captura, recuperação, acesso, uso e manutenção do conhecimento, sendo representados por:

- serviços que permitem a captura dos três tipos de itens de conhecimento;
- uma ferramenta que apóia o fluxo de trabalho envolvido na aprovação de uma lição aprendida;
- um sistema de gerência de configuração;
- ferramentas para instanciação das ontologias;
- uma serviço que permite a busca de qualquer item de conhecimento;
- um serviço que informa quem utilizou um determinado conhecimento, além de quando e quão freqüentemente ele tem sido utilizado;
- um serviço para avaliação, pelo usuário, dos itens de conhecimento utilizados; e
- um serviço de exclusão de itens de conhecimento e configuração de agentes de software para apoiar a manutenção da memória organizacional, lembrando o

gerente de conhecimento de efetuar a manutenção ou sugerindo itens de conhecimento a serem excluídos.

Já os serviços de distribuição de conhecimento são específicos das ferramentas, ou seja, são implementados por cada ferramenta através do uso da tecnologia de agentes. Um agente geral monitora as ações dos usuários e verifica quando uma ferramenta com gerência de conhecimento é iniciada. Quando isto ocorre, um agente específico para esta ferramenta começa a monitorar as ações do usuário para identificar as suas necessidades de conhecimento e recuperar experiências passadas similares. Um *framework* é fornecido para auxiliar o desenvolvedor de uma ferramenta na definição e configuração do respectivo agente específico. *ControlQ* e *ODEd* são exemplos de ferramentas para as quais foram definidos e configurados agentes específicos para a distribuição de conhecimento [87,155,156].

Por fim, a arquitetura do ambiente possui dois níveis: o nível base ou de aplicação, que contém as classes que modelam os objetos que implementam o apoio a alguma atividade de engenharia de software, e o nível meta ou do conhecimento, que contém classes derivadas diretamente das ontologias e que descrevem o conhecimento sobre os objetos no nível de aplicação. As classes no nível de aplicação também são construídas com base nas ontologias, mas novas classes, associações, atributos e operações são definidos de acordo com as decisões de projeto.

4.8 Outros Projetos

Além dos dois projetos anteriores, que têm sido conduzidos em paralelo com este trabalho, e dos projetos descritos por KOBIALKA [124], FALBO [123], OLIVEIRA [20] e ARBAOUI *et al.* [125], outros projetos têm sido conduzidos na área de ADS:

- *ASEE (Agile Software Engineering Environment)*, composto de um ADS Centrado em Processo e de um ambiente de compartilhamento de informações baseado na *Web*. *ASEE* visa a rápida distribuição de produtos de software e a rápida adaptação a mudanças de requisitos, promove uma fábrica de software virtual através da Internet e tem sido usado para desenvolver sistemas de software de comunicação [157];
- *ADDD (A Development environment for Dynamic Documents)*, um ADS Centrado em Processo que integra gerência de tarefa com gerência de configuração e cuja máquina de processo é baseada no paradigma de *triggers*, possibilitando instalação incremental do suporte ao processo, adaptação deste suporte de acordo os requisitos da tarefa específica, modificação do processo durante a execução e utilização de políticas para orientar e restringir o processo [124];

- PROSYT (*Process Support System*), um ADS Centrado em Processo especialmente concebido para apoiar processos de software distribuídos, que adota uma arquitetura baseada em código móvel e eventos, possui uma linguagem de modelagem de processo baseada em artefato e oferece flexibilidade na execução do modelo de processo [133];
- ASTRAL, um ADS que é um conjunto integrado de ferramentas de análise e projeto baseado na linguagem ASTRAL, uma linguagem de especificação formal para sistemas de tempo real que permite a especificação modular de sistemas complexos. O ADS fornece funcionalidades para reduzir erros e facilitar a manutenção e a reutilização de especificações [158];
- CHIME (*Columbia Hypermedia IMersion Environment*), um ADS baseado em meta-dados cujo objetivo é apoiar, através do uso da tecnologia de ambientes virtuais, grandes projetos de desenvolvimento de software, que, freqüentemente, têm membros dispersos geográfica e temporariamente [159];
- *Reusable Software Engineering Environment*, um ADS que apóia as atividades de engenharia de domínio (modelagem, projeto e implementação de componentes) bem como as atividades de configuração do sistema desejado (geração da especificação, composição da arquitetura e geração da aplicação executável), sendo composto de um ambiente de modelagem de domínio e um ambiente de configuração e programação distribuída [160];
- *Odyssey*, um ADS que apóia o desenvolvimento de software baseado em componentes, contemplando desde a elaboração dos modelos conceituais até a implementação dos componentes. Modelos conceituais, arquiteturais e de implementação são elaborados para domínios previamente selecionados e armazenados utilizando tecnologia de mediadores, que apóia heterogeneidade e distribuição [161];
- TDE (*Telecom Design Environment*), um ambiente que possibilita gerência das informações do projeto; fornece ferramentas gráficas que apóiam o desenvolvimento de software e permite desenvolvedores compartilharem o mesmo espaço de trabalho e trabalhar sobre os mesmos artefatos, tendo sido adaptado para fornecer modelos de processos enriquecidos com orientação, mas sem oferecer suporte à execução do processo e ao controle do seu progresso [162];
- VPP (*Visual Programming Platform*), um ADS voltado para o desenvolvimento de software para instrumentos virtuais, onde um instrumento virtual é um instrumento que contém hardware e software com algumas funcionalidades, tais como processamento e apresentação de dados e controles inteligentes. O ambiente

permite que o usuário graficamente crie diagramas da aplicação a partir de uma biblioteca de componentes [163];

- APSEE, uma infra-estrutura integrada para automação de processos de software, composta de componentes que armazenam informações relativas ao meta-modelo, mecanismos para gerência de processo, e componentes de visualização e interação. A partir da especificação formal APSEE, o código fonte básico do ambiente é gerado. As principais contribuições são a flexibilidade na execução dos processos, permitindo a modificação dinâmica dos mesmos, e o grau de automação das diferentes fases do ciclo de vida [164].

4.9 Conclusão

Neste capítulo, discutiu-se a evolução dos ambientes de desenvolvimento de software para apoiar a modelagem e a execução dos processos de software como também para apoiar os desenvolvedores de software na execução de suas atividades, fornecendo-lhes conhecimento sobre o domínio da aplicação. A tendência atual é apoiar a gerência de conhecimento num contexto mais amplo. Neste sentido, dois projetos recentes foram discutidos: a evolução do sistema MILOS [94,135] e a construção do ambiente ODE [87,153]. Os dois próximos capítulos abordam a definição e construção de Ambientes de Desenvolvimento de Software Orientados a Organização [21,165,166], tornando possível uma análise comparativa das propostas na conclusão deste trabalho.

Capítulo 5 - Ambientes de Desenvolvimento de Software Orientados à Organização

5.1 Introdução

Este capítulo define Ambientes de Desenvolvimento de Software Orientados a Organização (ADSOrg) e um modelo que especifica a infra-estrutura necessária para que tais ambientes contemplem os seus requisitos.

O conceito de ADSOrg estende o conceito de ADSOD [20,136], apresentado no capítulo 4, a partir da constatação de que conhecimento sobre o domínio de aplicação é apenas um dos tipos de conhecimento requeridos no desenvolvimento e na manutenção de software.

A ênfase do modelo aqui proposto é a definição de componentes que contemplem, explicitamente, os grupos de conhecimento mencionados nos requisitos definidos para ADSOrg, estabelecendo uma ontologia, o uso de alguns modelos e uma linguagem para representação dos conhecimentos considerados mais básicos: o conhecimento sobre a própria organização e sobre seus processos.

5.2 Conceito de ADSOrg

No capítulo 1, foi constatado que desenvolvedores de software lidam de forma intensa com diferentes tipos de conhecimento ao longo dos processos de software e que, por isso, maior atenção precisa ser dada à produção, armazenamento, compartilhamento e uso dos conhecimentos relevantes neste contexto. Uma abordagem natural para tratar esta questão é a introdução de Gerência do Conhecimento (discutida nos capítulos 2 e 3) nos Ambientes de Desenvolvimento de Software (discutidos no capítulo 4). A compatibilidade entre as abordagens pode ser percebida pelos pontos comuns existentes entre a Gerência do Conhecimento Orientada a Processo e os ADS Centrados em Processo e pelos resultados dos primeiros esforços conduzidos neste sentido [21,94,153].

Desta forma, os Ambientes de Desenvolvimento de Software Orientados a Organização (ADSOrg) [21] foram definidos como sendo ADS que apóiam a Gerência do Conhecimento ao longo dos processos de desenvolvimento e manutenção de software. Estes ambientes têm como objetivo: (a) apoiar os desenvolvedores de software na execução de suas atividades, fornecendo todo o conhecimento que tenha sido capturado e acumulado pela organização por sua importância para o

desenvolvimento e a manutenção de software, e (b) apoiar o aprendizado organizacional em Engenharia de Software a partir do aprendizado dos desenvolvedores da organização nos projetos de software específicos. A finalidade é evitar erros já cometidos e possibilitar a reutilização de soluções já aprovadas na execução de tarefas similares, buscando melhorar a produtividade e a qualidade bem como diminuir custos.

5.3 Tipos de ADSOrg

Em termos gerais, existem dois tipos de organizações que desenvolvem e mantêm software: Organizações de *Software*, nas quais o desenvolvimento e a manutenção de software são atividades de negócio, e Organizações com outro Tipo de Negócio, nas quais o desenvolvimento e a manutenção de software são atividades de suporte ao negócio [21].

Quando o desenvolvimento e a manutenção de software são atividades de negócio, a meta das organizações é uma combinação de afinidade com o cliente, excelência operacional e liderança de produto, na qual cada um destes fatores deve ser alcançado em algum grau, de acordo com as prioridades organizacionais¹. No entanto, independente da meta específica, o objetivo é sempre assegurar ou melhorar a posição atual da organização no mercado de software. Tais organizações podem precisar dominar vários processos, métodos, técnicas e ferramentas, tanto para atender às possíveis exigências dos clientes quanto para estabelecer o ambiente de desenvolvimento de software mais adequado para cada um de seus projetos. Além disto, ao longo de sua existência, podem acumular experiência de engenharia de software em diferentes domínios, sendo que o conhecimento sobre os vários domínios não pode ser obtido na própria organização, pelo menos no primeiro projeto envolvendo um determinado domínio.

Quando o desenvolvimento e a manutenção de software são atividades de suporte ao negócio da organização, a meta torna-se fornecer apoio adequado à realização plena das atividades de negócio ou criar uma maneira inovadora de realizá-las. Estas organizações necessitam dominar apenas os processos, métodos, técnicas e ferramentas utilizadas internamente e, ao longo de sua existência, acumulam experiência de engenharia de software no seu domínio de aplicação. O conhecimento do domínio pode ser obtido na própria organização.

¹ TREACY e WIERSMA [167] propõem a classificação das organizações de acordo com a prioridade dada a cada um dos três aspectos que agregam valor: afinidade com o cliente, excelência operacional e liderança de produto. KOUWENHOVEN [53] defende que estes aspectos podem ser combinados para estabelecer as metas estratégicas das organizações.

Uma vez que se pretende apoiar as atividades de desenvolvimento e manutenção de software em ambos os tipos de organização, foi considerada adequada a definição de dois tipos de ADSOrg: um para Organizações de Software e outro para Organizações com outro Tipo de Negócio, de forma a assegurar que as particularidades do desenvolvimento e manutenção de software em cada tipo de organização sejam contempladas.

5.4 Requisitos dos ADSOrg

Nesta seção, primeiro são apresentados os requisitos comuns a ambos os tipos de ADSOrg e, então, são discutidos os requisitos que são específicos para cada tipo.

➤ Requisitos Comuns

Independente do tipo específico, ADSOrg devem satisfazer os seguintes requisitos [21,165]: (i) armazenar e fornecer para as equipes de projeto um modelo da organização, contemplando estrutura, processos e capital intelectual, (ii) armazenar conhecimento especializado sobre o desenvolvimento e a manutenção de software na organização e fornecer este conhecimento para as equipes de projeto quando necessário, além de (iii) apoiar a contínua evolução do conhecimento armazenado no ambiente.

O modelo de uma organização contempla a sua estrutura, os seus processos e os conhecimentos, habilidades e experiências requeridos e possuídos ao longo desta estrutura e destes processos. A disponibilidade deste modelo para as equipes de software (requisito (i)) pode contribuir para o aumento da produtividade, na medida em que agiliza a localização de especialistas que podem trazer contribuições para o projeto e serve de base para levantamentos de requisitos relacionados aos processos organizacionais. A localização de especialistas é muito importante quando o conhecimento necessário não está disponível no ADSOrg ou quando, apesar de disponível, a sua aplicação requer entendimento mais profundo. A atividade de levantamento de requisitos também ganha em qualidade quando pode partir de um modelo de processos continuamente utilizado e validado.

Para atender ao requisito (ii), um ADSOrg precisa conter o conhecimento sobre as atividades de desenvolvimento e manutenção de software que são realizadas na organização independentemente de cliente ou projeto específico, o que constitui o processo de software da organização, também chamado de processo padrão [141]. Além disso, um ADSOrg deve armazenar a experiência de desenvolvimento e manutenção de software adquirida pela organização, o que pode envolver normas e diretrizes; últimas notícias relativas às tecnologias utilizadas; melhores práticas com

relação à definição de processos e à utilização de métodos e de linguagens de programação, com exemplos de projetos que as utilizaram; análise de ferramentas; produtos intermediários (propostas, planos, modelos, unidades de software executáveis) com potencial para serem reutilizados; roteiros de documentos; relatos de lições aprendidas em projetos anteriores; e medidas de desempenho organizacional. Sempre que pertinente, cada item de conhecimento armazenado no ambiente deve ser acompanhado da identificação de especialistas internos e materiais de referência que possam fornecer algum tipo de orientação adicional. Conhecimento sobre desenvolvimento e manutenção de software adquirido pela organização é fundamental para as atividades iniciais de um projeto, em que são elaborados a Proposta de Fornecimento e o Plano do Projeto, pois transforma estas atividades, que são centradas no conhecimento e na experiência do gerente do projeto, em atividades centradas no conhecimento e na experiência da organização, obviamente, acumulados a partir do conhecimento e da experiência de seus vários gerentes de projeto ao longo do tempo. Atividades de análise, projeto, implementação, teste e implantação podem ser apoiadas por notícias relativas às tecnologias utilizadas em cada atividade, melhores práticas em relação aos métodos selecionados, reutilização de produtos intermediários gerados por outros projetos, entre outros.

Uma questão importante é o fato do conhecimento de uma organização estar em constante evolução. Experiências relevantes para o desenvolvimento ou manutenção de software podem ser adquiridas a cada projeto, podendo conduzir à melhoria dos processos de software da organização, ao refinamento das bases de conhecimento de melhores práticas, ao acréscimo de produtos intermediários com alto potencial de reutilização, entre outros. Além disso, o modelo da organização também sofre alterações ao longo do tempo. Neste contexto, é fundamental que os ADSOrg apoiem a atualização do conhecimento organizacional fornecido pelos mesmos (requisito (iii)).

➤ **Requisitos de ADSOrg para Organizações de Software**

Organizações de Software normalmente possuem clientes em diversos domínios de negócio. A disponibilidade de conhecimento (mesmo que somente parcial) sobre os clientes e sobre os seus respectivos domínios representa uma vantagem estratégica na competição por novos projetos. Uma situação a ser evitada é a perda de conhecimento acumulado sobre um domínio e/ou sobre um cliente, uma vez que é comum o surgimento de outros projetos envolvendo o mesmo domínio e/ou o mesmo cliente. Em geral, uma *Organização de Software* monta equipes especializadas em determinados clientes, domínios e/ou tecnologias, mas dificuldades podem surgir devido a problemas como a rotatividade de pessoal e a impossibilidade de dispor de

pessoal já alocado em outro projeto. Desta forma, a construção de um modelo da organização cliente e a representação do conhecimento sobre o seu domínio de negócio podem ser úteis, mesmo que este modelo e representação sejam parciais. Assim, os requisitos dos ADSOrg para *Organizações de Software*, além dos requisitos comuns já discutidos, são [21,165]: (i) armazenar e fornecer modelos das organizações clientes para as equipes de projeto, contemplando estrutura, processos e capital intelectual, bem como (ii) armazenar conhecimento sobre os domínios de negócio dos clientes e fornecer este conhecimento para as equipes de software ao longo dos projetos relacionados a cada cliente.

Conhecimento sobre a organização cliente e sobre o seu domínio de negócio é fundamental nas atividades iniciais de um projeto de software, pois estas atividades também exigem uma visão global da organização cliente e experiência de desenvolvimento de software no seu domínio de negócio. No decorrer do projeto, conhecimento sobre o domínio é muito útil. Nas atividades de análise e de projeto, por exemplo, serve como ponto de partida para o levantamento de requisitos e para a modelagem dos dados do sistema [20]. Conhecimento sobre a organização cliente também pode ser útil no decorrer de um projeto, pois o conhecimento do domínio não elimina a interação com a organização cliente e o projeto pode ter como finalidade o apoio, controle ou automatização de um ou mais processos da organização cliente.

➤ **Requisitos de ADSOrg para Organizações com outro Tipo de Negócio**

Organizações com outro Tipo de Negócio desenvolvem e mantêm sistemas de software para uso próprio. Sendo assim, os requisitos de um ADSOrg neste contexto são os requisitos comuns já discutidos, acrescidos apenas dos requisitos: armazenar conhecimento sobre os domínios de negócio da organização e fornecer este conhecimento para as equipes de software ao longo dos projetos da organização. Estes requisitos são similares ao que caracterizam um ADSOD [20,136], onde o domínio da aplicação é, na verdade, um domínio de negócio da organização.

5.5 Modelo de ADSOrg

A figura 5.1 fornece uma visão macroscópica dos componentes de um ADSOrg. O *Repositório da Organização* e os *Serviços/Ferramentas de Gerência do Conhecimento* compõem a *Infra-estrutura de Gerência do Conhecimento*, que é essencial para um ADSOrg. Os *Serviços/Ferramentas de Gerência do Conhecimento* podem ser acessados diretamente por gerentes de conhecimento ou desenvolvedores de software, podendo também ser acessados através dos *Serviços/Ferramentas de*

Engenharia de Software, que fornecem acesso orientado a processo ao repositório da organização.

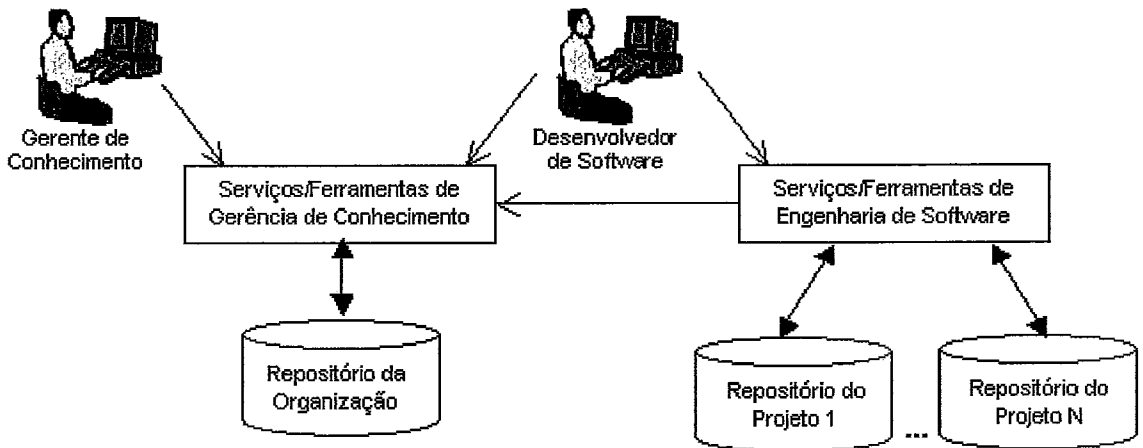


Figura 5.1 – Visão Macroscópica dos Componentes de um ADSOrg

As figuras 5.2a e 5.2b detalham o *Repositório da Organização* para os dois tipos de ADSOrg definidos.

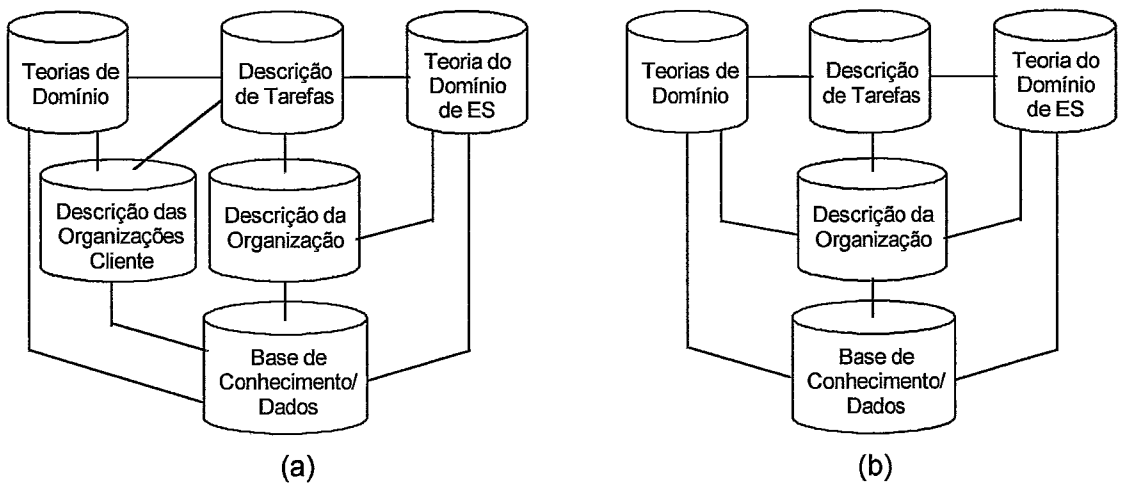


Figura 5.2 – Repositório da Organização: ADSOrg para Organizações de Software (a) e ADSOrg para Organizações com Outro Tipo de Negócio (b)

O componente *Descrição de Tarefas* contém a descrição de tarefas genéricas, comuns a diferentes domínios e organizações, como são, por exemplo, as tarefas de reservar e diagnosticar. Como definido por ZLOT [137] e OLIVEIRA *et al.* [136], a descrição de uma tarefa é formada por uma descrição de alto nível, uma ontologia dos conceitos associados com a tarefa (*Ontologia da Tarefa*), sua decomposição em termos de sub-tarefas ou as inferências necessárias a sua solução e um conjunto de referências bibliográficas. O objetivo de descrever tarefas genéricas é obter um conjunto de conhecimento independente de domínio e de organização, que apóie o desenvolvedor no entendimento de um problema a partir do entendimento das tarefas

que compõem o problema. Desta forma, o entendimento dos conceitos de um domínio de aplicação ou do domínio de Engenharia de Software pode ser facilitado pelo entendimento das tarefas em que os conceitos são utilizados e o entendimento dos processos de uma organização pode ser facilitado pelo entendimento das tarefas que os compõem.

O componente *Teorias de Domínio* organiza o conhecimento sobre os domínios de negócio, utilizando ontologias e identificando tarefas genéricas que são executadas no contexto dos domínios. Em ADSOrg para Organizações de Software (figura 5.2a), o conhecimento organizado refere-se aos domínios de negócio das organizações clientes e, em ADSOrg para Organizações com Outro Tipo de Negócio (figura 5.2b), o conhecimento organizado refere-se aos domínios de negócio da própria organização. Uma *Teoria de Domínio* é decomposta em subteorias, que, por sua vez, podem ser decompostas em novas subteorias ou conter uma ontologia e o mapeamento com as tarefas genéricas que aplicam os conceitos da ontologia. Uma *Ontologia de Domínio* define um vocabulário para representar o conhecimento sobre o domínio, expressando conceitos, propriedades, relações e restrições desse domínio. Além de promover o entendimento do domínio, esta ontologia é utilizada para orientar o registro e a atualização do capital intelectual da organização no que se refere aos conhecimentos e experiências relativos ao domínio.

O componente *Teoria do Domínio de ES* é similar ao componente *Teorias de Domínio*, sendo que o universo de discurso é a Engenharia de Software (ES). A finalidade da *Ontologia de ES* é fornecer um vocabulário comum para a comunicação entre os desenvolvedores e para a representação de, ou referência a, conhecimentos relacionados ao desenvolvimento e manutenção de software. Quando o capital intelectual da organização está sendo registrado ou atualizado, a indicação dos conhecimentos e experiências relativos à Engenharia de Software possuídos pelos funcionários podem ser definidos com base nos conceitos da *Ontologia de ES*. Além disso, itens de conhecimento podem ser definidos e associados com um ou mais conceitos da ontologia, o que possibilita a recuperação semântica dos mesmos.

O componente *Descrição da Organização* descreve a organização que utiliza o ADSOrg para desenvolver e manter software, estabelecendo, dentro do contexto da estrutura e dos processos organizacionais, quais as tarefas genéricas que são executadas e o conhecimento de Engenharia de Software que é requerido. Se a organização desenvolve software para uso próprio (*Organização com outro Tipo de Negócio*), este componente também estabelece qual conhecimento sobre os domínios de negócio é requerido ao longo de sua estrutura e de seus processos organizacionais. O mapa das competências disponíveis na organização também faz

parte do componente *Descrição da Organização* e especifica quais são os conhecimentos, habilidades e experiências possuídos por cada funcionário e em que grau estes conhecimentos, habilidades e experiências são possuídos. Os modelos dos processos organizacionais permitem especificar o contexto em que um item de conhecimento foi criado e o contexto de aplicação pretendido para o mesmo.

Descrição das Organizações Cliente é um componente específico dos ADSOrg para Organizações de Software (figura 5.2a), similar ao componente *Descrição da Organização*, mas que contém a descrição das organizações cliente da organização de software. Esta descrição também referencia as tarefas genéricas que são executadas e o conhecimento de domínio que é utilizado ao longo da estrutura e dos processos organizacionais.

Neste contexto, é fundamental dispor de uma *Ontologia de Organização* que defina um vocabulário comum para a descrição de qualquer organização. Este vocabulário inclui conceitos, propriedades, relações e restrições que contemplam os diferentes aspectos de um modelo organizacional.

O componente *Bases de Conhecimento/Dados* armazena o conhecimento e os dados relevantes para a organização que foram obtidos ao longo de seus vários projetos de software. Como já mencionado, os itens de conhecimento/dados, ao serem armazenados, devem ser associados a termos das ontologias existentes no ambiente, de forma que possam ser recuperados, também, a partir da exploração e/ou seleção dos termos das ontologias. O processo de software da organização, seus processos especializados e as lições aprendidas pela organização são exemplos de itens de conhecimento/dados armazenados neste componente.

Os *Serviços e Ferramentas de Gerência do Conhecimento* formam, em conjunto, um componente cujo objetivo é facilitar o armazenamento de dados, conhecimentos e experiências no repositório da organização (memória organizacional provida pelo ambiente), além de apoiar a constante disseminação e atualização dos mesmos.

Entre os *Serviços e Ferramentas de Gerência do Conhecimento*, tem-se:

- Ferramenta para gerência do conhecimento armazenado no componente *Descrição de Tarefas*, possibilitando a descrição de tarefas, a atualização destas descrições e a localização das referências a uma determinada tarefa no repositório da organização;
- Serviço para visualização do conhecimento referente a uma tarefa a partir de outras ferramentas do ambiente;
- Ferramenta para gerência do conhecimento armazenado nos componentes *Teorias de Domínio e Teoria do Domínio de ES*, ou seja, uma ferramenta para definição, atualização, extensão e instanciação de Teorias de Domínio, também capaz de

localizar as referências a um determinado conceito ou instância de conceito no repositório da organização;

- Ferramenta para visualização de uma Teoria de Domínio que permita que o usuário explore a estrutura da teoria e, conseqüentemente, a estrutura das ontologias subjacentes;
- Serviço para visualização do conhecimento referente a um conceito ou uma instância de um conceito a partir de outras ferramentas do ambiente, incluindo menção aos conceitos ou instâncias de conceitos relacionados;
- Ferramenta para descrição, atualização, extensão e visualização de estruturas organizacionais, incluindo a alocação de profissionais a essas estruturas;
- Ferramenta para descrição, atualização e visualização de processos organizacionais, permitindo acesso a conhecimento explícito manipulado pelos processos;
- Ferramenta para gerência de competências, que possibilite o registro e atualização das competências possuídas pelos profissionais de uma organização;
- Serviço de localização de profissionais com determinadas competências ou com o conjunto de competências mais próximo do solicitado;
- Ferramenta para definição e especialização de processos de software para a organização, a serem utilizados pelos gerentes de projeto na definição dos processos para os projetos específicos;
- Ferramentas para desenvolvimento (criação e/ou captura) de conhecimento ao longo dos projetos de software da organização;
- Ferramentas para análise e consolidação dos conhecimentos registrados no nível de projeto, de forma a gerar conhecimento disponível para os demais desenvolvedores da organização, o que envolve empacotamento apropriado do conhecimento;
- Ferramenta para recuperação e acesso a itens armazenados na *Base de Conhecimento/Dados* a partir de conceitos e relações das ontologias descritas no ambiente;
- Serviço para disseminação de conhecimento orientada a processo, ou seja, de acordo com a atividade que está sendo executada pelo desenvolvedor. Outros critérios podem ser utilizados para restringir ainda mais o conhecimento fornecido à necessidade do desenvolvedor e evitar sobrecarga de conhecimento;
- Serviço para avaliação do grau de utilidade do conhecimento acessado ao problema proposto pela atividade sendo executada, possibilitando a adição de

comentários que auxiliem na interpretação da avaliação realizada e na evolução do conhecimento e dos mecanismos de disseminação de conhecimento;

- Serviço para análise do uso dos itens de conhecimento fornecidos no *Repositório da Organização* de acordo com a frequência de uso, grau de utilidade e comentários gerais dos desenvolvedores, de forma a identificar itens de conhecimento candidatos à evolução ou à exclusão;
- Serviço para identificação de itens de conhecimento similares no *Repositório da Organização*, de forma oferecer oportunidades para consolidação dos mesmos, melhorando o desempenho dos ambientes.

Na figura 5.1, tem-se, ainda, os *Serviços e Ferramentas de Engenharia de Software* e os *Repositórios de Projeto*. Os *Serviços e Ferramentas de Engenharia de Software* têm como objetivo apoiar as atividades dos processos de desenvolvimento e manutenção de software, que envolvem atividades de construção, de gerência do projeto e de avaliação da qualidade. As *Ferramentas de Engenharia de Software* devem ser capazes de fornecer conhecimento organizacional relevante para o projeto no momento em que ele se faz necessário para os desenvolvedores, utilizando, para isto, a *Infra-estrutura de Gerência do Conhecimento*. Exemplos são uma ferramenta de apoio à definição de processos para projetos específicos a partir dos modelos que representam as orientações organizacionais; uma ferramenta que apóie o gerente na identificação dos riscos do projeto, fornecendo os riscos identificados em projetos similares da organização; uma ferramenta de modelagem capaz de reutilizar a definição e a estrutura de organização dos conceitos de uma Teoria do Domínio na construção de um modelo de dados ou de classes. Um *Repositório de Projeto* armazena todas as informações relativas ao projeto de software.

É importante ressaltar que a diferença no *Repositório da Organização* para os dois tipos de ADSOrg (figuras 5.2a e 5.2b) é conceitual, o que significa que é possível estabelecer uma infra-estrutura única para construção de ambos os tipos de ADSOrg. Como mencionado anteriormente, a diferença no nível conceitual teve como objetivo assegurar que os requisitos específicos de cada tipo de ADSOrg fossem contemplados pela Infra-estrutura de Gerência do Conhecimento proposta.

Outro aspecto a ser destacado é que ADSOrg são complexos por natureza, pois acrescentam Gerência do Conhecimento nos já complexos ADS, além de contemplarem uma amplo espectro de conhecimento. Desta forma, foi realizada uma pesquisa, descrita no próximo capítulo, com o intuito de orientar a incorporação de conteúdo e o desenvolvimento de serviços e ferramentas de forma evolutiva, e o

capítulo 7 descreve uma estratégia para a construção de ADSOrg baseada no conceito de meta-ambiente.

Além disso, a definição de *Teorias de Domínio* para domínios de negócio específicos já foi tratada por OLIVEIRA [20] e OLIVEIRA *et al.* [136] e depende da organização para qual será construído o ADSOrg e, a depender do tipo de organização, das suas organizações clientes. Já a *Teoria do Domínio de ES* não será abordada neste trabalho por ser bastante ampla, devendo ser construída de forma evolutiva e de acordo com as prioridades definidas pelos próprios desenvolvedores. Sendo assim, a ontologia de processo de software definida por FALBO [123] pode ser utilizada na versão inicial deste componente, podendo ser integradas, através de trabalhos futuros, outras ontologias já definidas, tais como as de manutenção de software definidas por KITCHENHAM *et al.* [168] e por DIAS [169] e a de qualidade de software definida por DUARTE [170].

No entanto, ainda restam aspectos fundamentais a serem tratados no âmbito do modelo proposto: a definição de uma ontologia de organização, que forneça um vocabulário genérico e reutilizável para descrição de diferentes organizações, e a modelagem de processos, englobando tanto os processos de software quanto os demais processos organizacionais, pois suporte ao processo de software é essencial em um ADSOrg e os demais processos são, em geral, os objetos da automatização dos projetos de software. Estes aspectos são abordados nas seções a seguir (seções 5.6 e 5.7).

5.6 Modelagem de Processos em ADSOrg

Dois tipos de processo são importantes no contexto dos ADSOrg: os processos de software, que têm a finalidade de orientar o desenvolvimento, reparo e melhoria de produtos de software; e os processos da organização e, quando pertinente, das suas organizações clientes que podem ser apoiados, controlados ou automatizados por produtos de software. Nos ADSOrg, os modelos de processos de software são prescritivos, ou seja, fornecem instruções de como os processos devem ser executados para que os objetivos pretendidos sejam atingidos; enquanto os modelos dos demais processos são descritivos, pois a finalidade é promover o entendimento desses processos e facilitar a comunicação, possibilitando uma visão compartilhada dos mesmos. Esta distinção resulta em tratamentos diferenciados para os modelos de processo no contexto dos ADSOrg.

Como visto no capítulo anterior, uma série de linguagens e formalismos para modelagem de processo de software têm sido propostos, mas a complexidade das linguagens existentes e a forte orientação à execução e à automação dos processos

dificultam a adoção dos mesmos na prática, onde o mais importante é a comunicação e entendimento dos processos. Recentemente, REIS *et al.* [134] propuseram uma linguagem de modelagem gráfica para descrever processos de software com o objetivo de apoiar a instanciação, considerando que modelagem, instanciação e execução são freqüentemente intercaladas no ciclo de vida do processo. O *Object Management Group*² (OMG) definiu um meta-modelo para engenharia de processos de software, que pode ser considerado uma UML (*Unified Modeling Language*) *Profile*³, utilizando estereótipos e diagramas UML básicos para fornecer diferentes perspectivas de um modelo de processo de software [171].

Na subseção 5.6.1, a modelagem de processos de software em ADSOrg é discutida, onde o foco não é a linguagem de modelagem utilizada, que deve ser a mais simples e intuitiva possível, mas como os modelos de processos de software podem ser definidos de forma sistemática e incremental, e utilizados para guiar os desenvolvedores nas atividades a serem executadas.

No que se refere à modelagem de processos organizacionais quaisquer, a *Rational Software* propôs uma UML *Profile* para modelagem de negócio [172] e muitos Sistemas de Gerência de *Workflow*⁴ e Ferramentas de Definição de Processos têm sido desenvolvidos e comercializados. Diante da variedade de produtos de *workflow* no mercado, muitas vezes voltados para necessidades específicas, a *Workflow Management Coalition*⁵ (WfMC) definiu um modelo de referência para promover a integração de produtos de *workflow* [173]. Este modelo inclui um meta-modelo para definição e intercâmbio de processos [132], que estabelece um conjunto básico de tipos de objeto para a troca de definições de processo simples e possibilita que outros tipos de objetos sejam adicionados por extensões específicas de vendedor. O *Ultimus Workflow Suite* e o ARIS *Easy Design* e ARIS *Toolset* são exemplos de produtos de *workflow* que utilizam linguagem de modelagem gráfica para definição de processos, sendo que o ARIS *Easy Design* permite a documentação e acesso de conhecimentos associados aos processos utilizando os objetos gráficos definidos em ALLWEYER [10]. No entanto, enquanto o *Ultimus Workflow Suite* não contempla a representação

² O OMG é uma organização internacional fundada em 1989 para promover a teoria e a prática da tecnologia orientada a objeto no desenvolvimento de software.

³ Uma UML Profile é uma variação da UML que utiliza os mecanismos de extensão da UML de maneira padronizada para um propósito específico.

⁴ Sistemas de Gerência de *Workflow* são sistemas que definem, criam e controlam a execução de *workflows*, através do uso de software capaz de interpretar a definição do processo, interagir com os participantes do *workflow* e, quando necessário, chamar ferramentas e aplicações baseadas na Tecnologia da Informação [173].

⁵ A WfMC é uma organização internacional fundada em 1993 que tem a missão de promover e desenvolver o uso de *workflow* através do estabelecimento de padrões de terminologia, interoperabilidade e conectividade entre produtos de *workflow*.

dos conhecimentos requeridos e produzidos ao longo dos processos, os produtos ARIS prevêem um conjunto de 34 objetos com desenhos, muitas vezes, bastante semelhantes e o ARIS *Easy Design* utiliza objetos com desenhos desnecessariamente complexos (por exemplo: pessoa/organização, função, conhecimento documentado).

Sendo assim, a subseção 5.6.2 apresenta uma linguagem gráfica para modelagem de processos organizacionais quaisquer, inclusive os de software, com fins descritivos. A linguagem difere das demais porque: (i) busca definir um conjunto mínimo de objetos, suficientes para representar os conceitos da ontologia de organização considerados pertinentes no contexto, as primitivas mencionadas em [132,174] e alguns conceitos que se mostraram necessários na prática; (ii) permite a representação dos conhecimentos requeridos e produzidos ao longo dos processos e (iii) utiliza objetos gráficos simples de desenhar. O resultado é uma linguagem fácil de usar, intuitiva, que produz modelos fáceis de serem lidos, visualmente agradáveis, além de compatíveis com o desejo de modelar o fluxo de conhecimento ao longo dos processos organizacionais.

5.6.1 Modelos de Processos de Software

EMAN [175] separa o conhecimento sobre processos de engenharia de software em dois níveis. O nível meta, chamado Engenharia de Processos de Software, aborda a definição, medição, gerência, modificação e melhoria de processos de engenharia de software. A definição de processos, foco desta subseção, engloba tipos de definições de processo, modelos de ciclo de vida, processos de ciclo de vida, notação para definições de processos, métodos para definição de processos e automação [175]. A abordagem para definição de processos proposta por ROCHA *et al.* [143] e detalhada em [144] combina, de maneira sistemática e incremental, a maioria dos elementos mencionados por EMAN [175]. A figura 5.3 representa, de forma esquemática, a abordagem, que foi revista neste trabalho quanto aos conhecimentos considerados relevantes em cada etapa da definição e quanto ao conceito de processo padrão.

A primeira etapa da abordagem consiste na definição de um modelo de processo padrão para a organização de acordo com modelos de processos de ciclo de vida propostos na literatura e as características da organização. Além disso, quando desejado pela organização, um modelo de maturidade e capacidade pode também ser considerado.

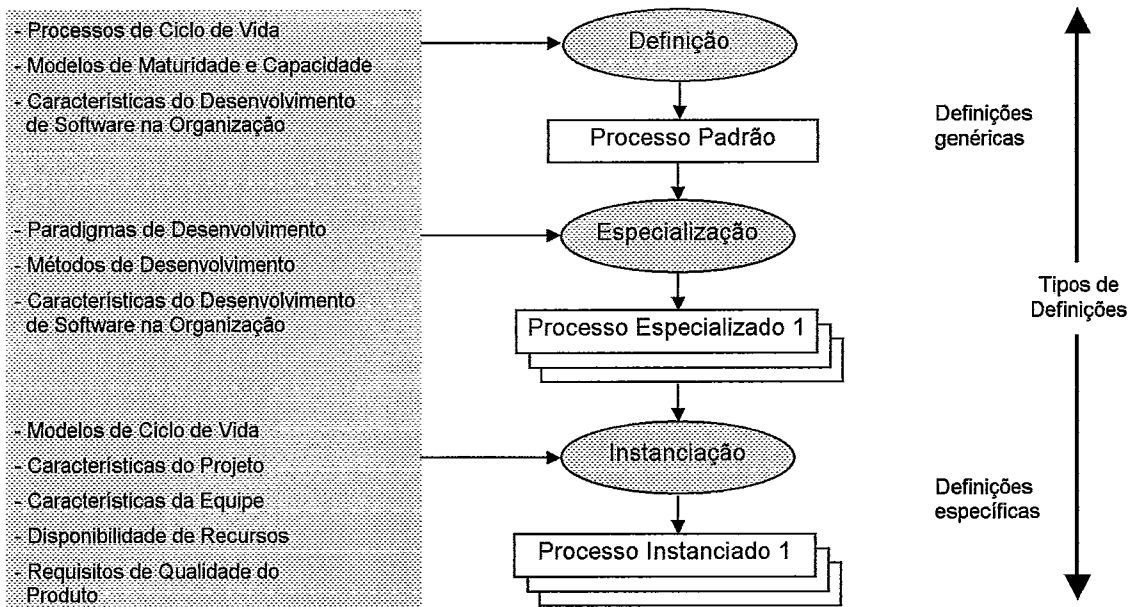


Figura 5.3 – Abordagem para Definição de Processos

As principais referências para a obtenção de modelos de processos de ciclo de vida são a ISO/IEC 12207 [140] e a ISO/IEC TR 15504 [176].

Dentre os modelos para avaliação da maturidade e capacidade, têm-se o CMM-SW (*Capability Maturity Model for Software*) [145] e o ISO/IEC TR 15504 [176]. No entanto, o *Software Engineering Institute* lançou, em dezembro de 2001, o CMMI (*CMM Integration*) em substituição ao CMM, distribuindo, em agosto de 2002, a versão do CMMI-SW (*CMMI for Software Engineering*) separada do CMMI-SE (*CMMI for System Engineering*) [177]. O novo modelo, que é, na verdade, um conjunto integrado de modelos, tem sido bem aceito pela comunidade.

A necessidade de considerar as características da organização na definição de seu processo padrão foi destacada por OLIVEIRA [20]. Exemplos de características relevantes nesta etapa são: o fato da organização desenvolver *software* para uso próprio, para comercialização ou sob encomenda; o fato da organização ser especializada no desenvolvimento de um tipo de *software* específico; as atividades realizadas nos projetos de *software* da organização que são consideradas específicas da organização; os problemas cruciais enfrentados pela organização em seus projetos de *software*, entre outras.

Segundo EMAM *et al.* [141], o processo padrão descreve os elementos fundamentais que *devem ser incorporados* em qualquer processo definido na organização e as relações entre esses elementos, como seqüência e interfaces. Ainda segundo o autor, a definição de um processo padrão deve considerar os objetivos do processo; identificar as atividades, papéis e responsabilidades; definir as entradas e

saídas, os pontos de controle e os registros de qualidade; e identificar *interfaces* externas e internas.

Considerando a importância de padronizar, mas também a de orientar, a definição de processo padrão adotada no contexto dos ADSOrg estabelece a descrição dos elementos *a serem considerados* na definição de cada processo definido na organização e também as relações entre esses elementos. Isto implica que nem todos os elementos previstos no processo padrão são sempre obrigatórios, pois alguns elementos podem não ser pertinentes em determinados contextos. A descrição de elementos não obrigatórios fornece maior orientação para os engenheiros de software responsáveis pela definição de processos, pois permite que a mesma descrição seja utilizada sempre que o elemento for considerado pertinente, colaborando com a padronização. Além disso, é mais fácil desconsiderar a descrição de uma atividade não obrigatória do que elaborar uma descrição para a mesma quando for necessário.

Na etapa da especialização, atividades do processo padrão são detalhadas de acordo com o paradigma e os métodos de desenvolvimento, que são escolhidos de acordo com as características do desenvolvimento de software na organização, tais como experiência dos desenvolvedores e tipos de software desenvolvidos. Exemplos de paradigma são o estruturado, o orientado a objetos e o baseado em conhecimento. O produto desta etapa são os modelos de processo especializados, que são importantes para apoiar o desenvolvimento e a manutenção de software na organização, refletindo o conhecimento e a experiência obtidos sobre o uso do paradigma e dos métodos. A depender do tipo de software, pode haver a necessidade de incluir atividades específicas para o tipo de software e/ou de especializar atividades já previstas.

Na etapa da instanciação, as características do projeto e da equipe, além da disponibilidade de recursos, são consideradas na escolha do modelo de ciclo de vida mais adequado ao projeto. Exemplos de tais modelos são: cascata, prototipagem descartável, prototipagem evolutiva, desenvolvimento incremental, espiral e baseado em reutilização. Atividades importantes no contexto do projeto podem ser incluídas nessa etapa, como, por exemplo, atividades que irão colaborar para que o produto apresente as características de qualidade desejadas [30,178]. Desta forma, o modelo de processo especializado mais adequado para o tipo de software a ser desenvolvido é instanciado e o modelo resultante é o que, de fato, será utilizado para execução do projeto.

A abordagem, além de orientar o engenheiro de software na utilização do conhecimento referente à definição de processos, gera produtos que constituem conhecimento especializado da organização em Engenharia de Software e promove o uso deste conhecimento na definição de processos para os projetos da organização. Portanto, apesar de não ter sido proposta com este objetivo, a abordagem é bastante adequada para apoiar a Gerência do Conhecimento em ADS, estando especificamente relacionada ao requisito (iii) dos ADSOrg (seção 5.4). Nesse contexto, o processo padrão da organização é utilizado para disseminar e evoluir conhecimento organizacional sobre como executar os processos de ciclo de vida.

Cabe ainda destacar que, enquanto o CMM-SW [145] e o ISO/IEC TR 15504 [176] utilizam o termo processo padrão para fazer referência a um conjunto integrado de processos de engenharia e gerência necessários ao desenvolvimento e manutenção de software, o CMMI (CMM *Integration*) [177] faz referência a um conjunto de processos padrão da organização, que contém as definições de processo que guiam todas as atividades da organização, cobrindo todos os processos necessários na organização e em seus projetos.

O tipo de notação e a notação específica utilizada para definição dos modelos de processo de software não são especialmente importantes no contexto dos ADSOrg, sendo apenas requerido, como já mencionado, que a notação seja simples e intuitiva e que os modelos de processo resultantes sejam capazes de, em conjunto com mecanismos de *interface* do ambiente, guiar os desenvolvedores nas atividades a serem executadas. A automação, que refere-se ao suporte automatizado à definição e execução de processos, é uma das responsabilidades do componente *Serviços/Ferramentas de Engenharia de Software* do modelo de ADSOrg (figura 5.1).

5.6.2 Linguagem para Modelagem de Processos Organizacionais

A linguagem proposta neste trabalho para modelagem de processos organizacionais é composta de elementos gráficos que podem ser do tipo área, objeto ou ligação, onde uma ligação estabelece uma relação entre dois objetos e uma área agrupa objetos, definindo um contexto para os mesmos. Objetos ainda permitem adornos, utilizados para representar explicitamente características dos objetos. A seguir, cada elemento da linguagem é apresentado.

Tabela 5.1 – Definição e Notação dos Objetos

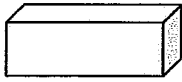










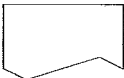
Objeto	Notação	Definição
Processo		Objeto referente ao conceito de mesmo nome definido na ontologia de organização (seção 5.7). <i>Atributos Especiais:</i> Origem (Interno, Externo)
Evento		Objeto que representa um acontecimento no ambiente que provoca o início ou fim de um processo. A notação é proveniente do produto comercial de <i>workflow ARIS ToolSet</i> .
Ator		Objeto que representa um pessoa, agente ou unidade organizacional. Estes conceitos encontram-se definidos na ontologia de organização. A notação foi utilizada por KRUCHTEN [179] para representação dos <i>workflows</i> básicos do <i>Rational Unified Process</i> .
Atividade		Objeto referente ao conceito de mesmo nome definido na ontologia de organização. A notação foi utilizada por KRUCHTEN [179] para representação dos <i>workflows</i> básicos do <i>Rational Unified Process</i> . <i>Atributos Especiais:</i> Origem (Interna, Externa) Granularidade (Elementar ou Composta)
Estado Inicial		Objeto puramente notacional, proveniente dos diagramas de estado e que indica onde é iniciado o fluxo de atividades que definem um processo ou uma atividade composta
Estado Final		Objeto puramente notacional, proveniente dos diagramas de estado e que indica onde é encerrado o fluxo de atividades que definem um processo ou uma atividade composta
Conhecimento Explícito		Objeto que representa um conhecimento que pode ser expresso em palavras e números e ser facilmente transmitido e compartilhado. A notação foi proposta por ALLWEYER [10].
Conhecimento Implícito		Objeto que representa um conhecimento que é altamente pessoal e difícil de formalizar, o que o torna também difícil de ser compartilhado. A notação foi proposta por ALLWEYER [10].
Comunicação		Objeto que representa a comunicação de dados ou informações a partir da, ou para a, execução de uma atividade. A comunicação pode ser verbal ou escrita e exemplos são e-mail e fax.
Repositório (Meio Magnético)		Objeto que representa um meio magnético para o armazenamento de dados e informações. A notação é proveniente do produto comercial de <i>workflow ARIS ToolSet</i> .
Arquivo (Local Físico)		Objeto que representa um local físico para armazenamento de documentos e comunicações escritas.
Documento		Objeto referente ao conceito de mesmo nome definido na ontologia de organização. A notação é proveniente do produto comercial de <i>workflow ARIS ToolSet</i> .

Tabela 5.1 – Definição e Notação dos Objetos (continuação)





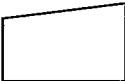
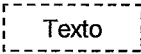
Objeto	Notação	Definição
Componente de Hardware		Objeto referente ao conceito de mesmo nome definido na ontologia de organização. A notação é baseada na notação de componente da UML.
Componente de Software		Objeto referente ao conceito de mesmo nome definido na ontologia de organização. A notação é baseada na notação de componente da UML.
Peça		Objeto referente ao conceito de mesmo nome definido na ontologia de organização.
Matéria-Prima		Objeto referente ao conceito de mesmo nome definido na ontologia de organização.
Bem		Objeto referente ao conceito de mesmo nome definido na ontologia de organização. A notação fornecida pode ser substituída por uma mais significativa para o objeto específico do modelo como, por exemplo, o logotipo do software. <i>Atributos Especiais:</i> Tipo (Usufruto, Software, Hardware e Equipamento de Produção)
Nota Explicativa		Objeto que permite que notas explicativas sejam adicionadas ao modelo. <i>Atributos Especiais:</i> Texto

Tabela 5.2 – Definição e Notação dos Adornos

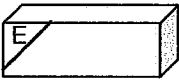
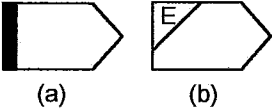
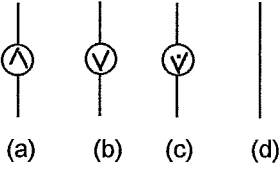

Objeto	Notação com Adornos	Definição dos Adornos
Processo		Adorno que indica que o processo é externo, ou seja, que é executado por outra organização.
Atividade		(a) Adorno que indica que a atividade é composta, o que significa que ela pode ser decomposta em sub-atividades; (b) Adorno que indica que a atividade é externa, ou seja, que é executada por outra organização.
Operação Lógica		(a) Adorno que indica a operação lógica E; (b) Adorno que indica a operação lógica OU; (c) Adorno que indica a operação lógica OU Exclusivo; (d) Notação sem adorno, que é utilizada como elemento de junção após uso de uma operação lógica que atua como elemento de divisão.
Conhecimento Explícito		Adorno que indica que foi especificado um caminho para acesso ao conhecimento disponível em meio magnético. Este adorno só deve ser utilizado se a visualização do modelo for apoiada por uma ferramenta de software que permita o acesso ao conhecimento. <i>Atributos Especiais:</i> Localização do Arquivo

Tabela 5.3 – Definição e Notação das Áreas

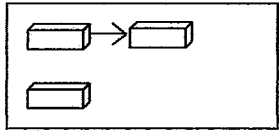
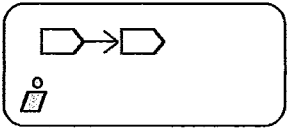
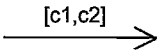



Objeto	Notação	Definição
Grupo de Processos		Área que agrupa processos relacionados.
Área de Ator		Área que agrupa atividades executadas por um ator ou grupo de atores. O ator ou o grupo de atores também precisa estar contido na área.

Tabela 5.4 – Definição e Notação das Ligações

Objeto	Notação	Definição
Fluxo de Controle		Ligação que indica a passagem de controle do objeto origem para o objeto destino. O c1 e o c2 indicados na notação são os rótulos das condições estabelecidas para que a passagem de controle ocorra. <i>Atributo Especial:</i> Condição, formada por rótulo e descrição
Fluxo de Entrada/Saída		Ligação que estabelece um insumo (se o fluxo é de entrada) ou um produto de uma atividade (se o fluxo é de saída). Quando o objeto de origem ou destino é um armazenador (repositório ou arquivo), a notação pode incluir os rótulos das informações trafegadas, existindo, então, um atributo especial. <i>Atributo Especial:</i> Informação, formada por rótulo e descrição
Ligação Não Direcionada		Ligação que não indica passagem de controle nem estabelece insumos e produtos para uma atividade, sendo utilizada para conectar bens de produção (software, hardware e equipamentos) utilizados como recursos para execução das atividades e para conectar eventos que atuam sobre processos, provocando o seu início ou fim. No segundo caso, um atributo especial é definido. <i>Atributo Especial:</i> Papel do Evento (Iniciador, Terminador)
Ligação para Nota Explicativa		Ligação que estabelece que uma nota explicativa é referente a um elemento do modelo.

A regras para composição dos elementos gráficos da linguagem são:

- Um Grupo Processo agrupa um⁶ ou mais Processos e um Processo pode estar ou não em um Grupo Processo;
- Uma Área Ator agrupa um ou mais Atores e um Ator sempre está contido em uma Área Ator;
- Uma Área Ator agrupa uma⁶ ou mais Atividades e uma Atividade pode estar ou não contida em uma Área Ator;
- Um Fluxo de Controle pode conectar um Processo a outro Processo ou a uma Operação Lógica;

⁶ Zero é admitido enquanto o modelo está sendo elaborado.

- Um Fluxo de Controle pode conectar uma Atividade a outra Atividade, a uma Operação Lógica, a um Estado Inicial ou a um Estado Final, sendo que, quando a ligação envolve um dos dois últimos objetos, o Estado Inicial é obrigatoriamente a origem e o Estado Final é obrigatoriamente o destino;
- Tanto Processos quanto Atividades admitem apenas um Fluxo de Controle com destino na atividade/processo;
- Um Fluxo de Controle pode conectar uma Operação Lógica a um Processo, uma atividade, outra Operação Lógica, um Estado Inicial ou um Estado Final. Quando a ligação envolve um dos dois últimos objetos, o Estado Inicial é obrigatoriamente a origem e o Estado Final é obrigatoriamente o destino;
- Uma Operação Lógica implementa uma primitiva de Junção (*Join*) ou de Divisão (*Split*), o que significa, respectivamente, que apenas um Fluxo de Controle tem origem na Operação Lógica ou apenas um Fluxo de Controle tem como destino a Operação Lógica;
- Uma Ligação Não Direcionada pode conectar um Processo e um Evento ou uma Atividade e um Bem;
- Através de Ligações Não Direcionadas, um Processo pode estar conectado a zero ou mais Eventos, e um Evento pode estar conectado a um⁶ ou mais Processos;
- Através de Ligações Não Direcionadas, uma Atividade pode estar conectada a zero ou mais Bens, e um Bem pode estar conectado a zero ou mais Atividades;
- Um Fluxo de Entrada/Saída pode conectar uma Atividade a um Conhecimento Implícito, a um Conhecimento Explícito, a uma Comunicação, a um Repositório, a um Arquivo, a um Componente de Hardware ou de Software, a uma Peça, a um Documento, a um Bem ou a uma Matéria-Prima, sendo que, quando a ligação envolve Matéria-Prima, este objeto é obrigatoriamente a origem;
- Através de Fluxos de Entrada/Saída, uma Atividade pode estar conectada a zero ou mais dos elementos pertinentes e cada um desses elementos pode estar conectado a uma⁷ ou mais Atividades;
- Uma Ligação para Nota Explicativa pode conectar qualquer elemento da linguagem a uma Nota Explicativa;
- Através de Ligações para Nota Explicativa, uma Nota Explicativa pode estar conectada a zero ou mais elementos da linguagem e um elemento da linguagem pode estar conectado a zero ou mais Notas Explicativas.

⁷ Zero é admitido enquanto o modelo está sendo elaborado. Além disso, Bens podem existir no modelo sem estar associados a Atividades através de Fluxos de Entrada/Saída.

Algumas considerações sobre o objeto Operação Lógica ainda são oportunas. Com o objetivo de modelar *workflows*, CHANG *et al.* [174] discutem como as primitivas *AND-Join*, *AND-Split*, *OR-Join*, *OR-Split*, Iteração e Casualidade podem ser expressas utilizando o diagrama de atividade da UML (figura 5.4). No meta-modelo definido em [132], a descrição de uma atividade pode ser dividida em elemento de junção, corpo e elemento de divisão, onde o elemento de junção trata das restrições sobre as transições de entrada, o corpo pode conter uma repetição e o elemento de divisão trata das restrições sobre as transições de saída. No entanto, as restrições sobre as transições são especificadas através do uso das palavras-chave AND e XOR, sendo que um AND com condições no elemento de divisão pode ser considerado um OR. Já a linguagem proposta possibilita a representação das primitivas *AND-Join*, *AND-Split*, *OR-Join*, *OR-Split*, *XOR-Join* e *XOR-Split* (figura 5.5), pois também foi considerado importante representar o OR sem condições associadas. Um outro aspecto a ser destacado é a representação explícita do OR e do XOR, principalmente nas primitivas de junção⁸, de forma a possibilitar combinações de primitivas (por exemplo: [Atividade1 ou Atividade 2] e Atividade 3).

A linguagem para modelagem de processos proposta nesta subseção tem sido avaliada por estudantes e pesquisadores da linha de Engenharia de Software da COPPE/UFRJ, que a têm utilizado para modelar os processos apoiados pelas ferramentas desenvolvidas [27,28,29,30,31,32,33,34,35,36]. Além de estar sendo avaliada, a evolução da linguagem é inerente à estratégia adotada para a sua definição, segundo a qual o conjunto mínimo, inicialmente proposto, de elementos e possibilidades de conexão desses elementos deve evoluir a partir da necessidade prática dos modeladores. Portanto, um requisito importante para uma ferramenta de apoio à modelagem de processo baseada na linguagem é o de que ela seja flexível, oferecendo mecanismos para a evolução da linguagem.

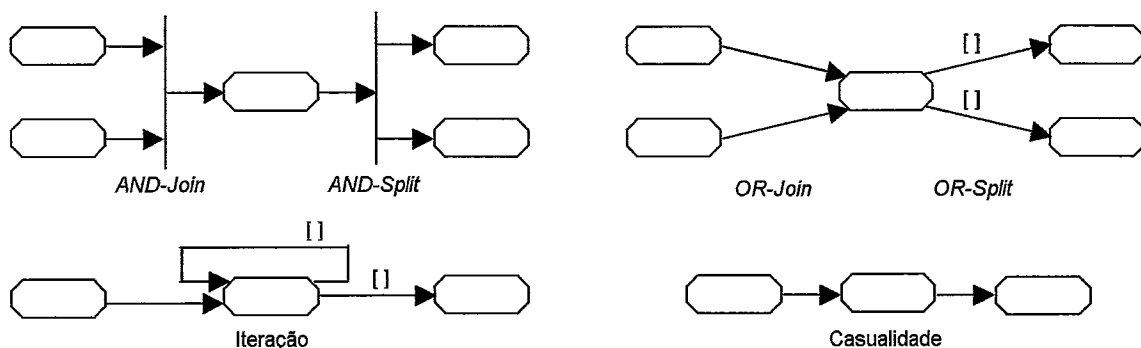


Figura 5.4 – Primitivas utilizando a Notação do Diagrama de Atividade UML

⁸ A primitiva *OR-Split* também pode ser representada explicitamente. A representação implícita visa apenas economizar espaço de desenho nos diagramas, o que pode ser crítico em modelos mais complexos.

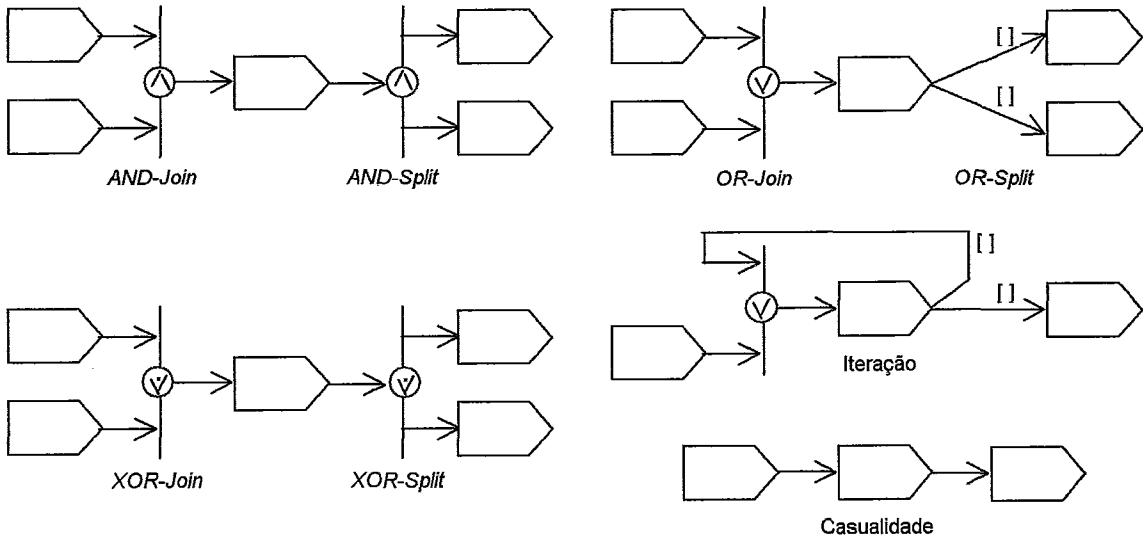


Figura 5.5 – Primitivas utilizando a Notação Proposta

5.7 Ontologia de Organização

Para construção da ontologia de organização, foi utilizado um processo definido com base nos trabalhos de USCHOLD e GRUNINGER [95], FALBO [123] e OLIVEIRA [20]. Os produtos gerados pelo processo são apresentados nesta seção.

5.7.1 Identificação do Propósito da Ontologia

A ontologia de organização tem como propósito fornecer um vocabulário comum que possa ser utilizado para representar conhecimento útil para os desenvolvedores de software sobre as organizações envolvidas em um projeto de software. A seguir, são apresentados os cenários de motivação⁹ que melhor representam a utilidade da ontologia para os ADSOrg, sendo que os 3 primeiros cenários são relacionados à construção dos ADSOrg e os 2 últimos são relacionados ao uso dos ADSOrg.

- Um ADSOrg deve fornecer conhecimento sobre uma ou mais organizações (conforme discutido nas seções 5.4 e 5.5) e este conhecimento precisa estar de alguma forma organizado. A ontologia de organização fornece a estrutura a ser utilizada para organizar o conhecimento e orienta a aquisição do conhecimento sobre as organizações;
- Construir um ADSOrg é uma atividade complexa. A estrutura definida pela ontologia de organização pode ser utilizada na construção de ADSOrg para diferentes organizações, o que favorece o desenvolvimento de ferramentas genéricas

⁹ Cenários de motivação são problemas ou situações que mostram a utilidade da ontologia [95].

baseadas na ontologia, diminuindo o esforço despendido na construção dos ADSOrg;

- A integração de ferramentas é uma preocupação constante em ADS (conforme mencionado no capítulo 4). Bases geradas a partir da estrutura definida pela ontologia de organização podem ser compartilhadas por ferramentas que manipulam conhecimentos relacionados à ontologia, promovendo a integração de dados;
- Um sistema que manipula conhecimento sobre a organização precisa ser desenvolvido. Exemplos são sistemas que apóiam processos organizacionais ou sistemas de informações gerenciais. Primeiro, os requisitos podem ser mais facilmente identificados se um vocabulário comum for utilizado pelas pessoas envolvidas na tarefa. Segundo, a estrutura para representação do conhecimento sobre a organização e o conhecimento já disponível podem ser aproveitados para compor uma primeira versão dos requisitos. Terceiro, é importante saber quem são pessoas na organização que podem contribuir com informações sobre o sistema. A ontologia de organização tem papel fundamental em todos os três itens;
- Um gerente de projeto está definindo uma equipe para um projeto e precisa encontrar os desenvolvedores mais adequados para compor a equipe de acordo com as características do projeto. Por outro lado, um desenvolvedor de software está precisando de orientação para execução de sua tarefa ou está querendo discutir idéias com um outro desenvolvedor e deseja saber quem na organização possui as competências necessárias para auxiliá-lo. Em ambos os casos, o conhecimento sobre a organização, obtido com base na ontologia de organização, pode auxiliar a identificação dos profissionais com as competências desejadas.

Em termos de definição do escopo da ontologia de organização, as seguintes questões gerais de competência¹⁰ (QG) foram formuladas:

- QG1: Como a organização é percebida em seu ambiente?
- QG2: Como a organização está estruturada?
- QG3: Quem são as pessoas que fazem parte da organização?
- QG4: Como a organização se comporta?
- QG5: Qual é a distribuição de autoridade e responsabilidade na organização?
- QG6: Como as competências desejadas e possuídas encontram-se distribuídas na organização?
- QG7: Quais são os objetivos estabelecidos para a organização?
- QG8: Como têm sido conduzidos os projetos da organização?

¹⁰ Questões de competência são questões que a ontologia deve possibilitar que sejam respondidas [95].

5.7.2 Definição e Formalização da Ontologia

A definição da ontologia de organização foi baseada em pesquisa bibliográfica sobre as ontologias relacionadas, elaboradas no contexto do projeto TOVE (*TO*ronto *V*irtual *E*nterprise) [180,181,182,183,184], do projeto *Enterprise* [185] e por COTA [186], em pesquisa bibliográfica na área de Administração de Empresas [24,187,188,189,190] e em entrevistas com especialistas da mesma área.

Inicialmente, as questões gerais de competência foram refinadas, dando origem às questões de competência listadas a seguir, que representam a especificação de requisitos da ontologia.

QG1: Como a organização é percebida em seu ambiente?

Q1.1: Qual é a missão da organização?

Q1.2: Quais são os domínios de conhecimento nos quais a organização atua?

Q1.3: Quais são os artefatos/serviços oferecidos pela organização?

Q1.4: Quais são as organizações clientes da organização?

QG2: Como a organização está estruturada?

Q2.1: Quais são os cargos existentes na organização?

Q2.2: Como a organização é decomposta em unidades organizacionais?

Q2.3: Como uma unidade organizacional é decomposta em posições?

Q2.4: Quais são as comissões existentes na organização?

QG3: Quem são as pessoas que fazem parte da organização?

Q3.1: Quais são as pessoas que possuem um determinado cargo na organização?

Q3.2: Quais são as pessoas que ocupam uma determinada posição na organização?

Q3.3: Quais são as pessoas que integram uma determinada comissão na organização?

QG4: Como a organização se comporta?

Q4.1: Quais são os processos executados na organização?

Q4.2: Quais são as normas seguidas por um determinado processo?

Q4.3: Como um determinado processo é decomposto em atividades?

Q4.4: Quais são os recursos e insumos necessários para executar uma determinada atividade?

Q4.5: Quais são os produtos resultantes da execução de uma determinada atividade?

Q4.6: Quais são as pessoas que estão alocadas para a execução de uma atividade?

Q4.7: Quais procedimentos podem ser utilizados para executar uma determinada atividade?

Q4.8: Quais são as ferramentas de software que automatizam um determinado procedimento?

QG5: Qual é a distribuição de autoridade e responsabilidade na organização?

Q5.1: A quais unidades organizacionais uma determinada unidade organizacional se reporta?

Q5.2: A quais unidades organizacionais uma determinada unidade organizacional assessora ou presta serviço?

Q5.3: A quais posições uma determinada posição se reporta?

Q5.4: A quais posições uma determinada posição assessora ou presta serviço?

Q5.5: Quais são as atividades que são atribuição de uma determinada posição ou cargo?

QG6: Como as competências desejadas e possuídas encontram-se distribuídas na organização?

Q6.1: Quais pessoas na organização possuem uma determinada competência?

Q6.2: Em quais atividades da organização uma determinada competência é requerida?

Q6.3: Em quais cargos da organização uma determinada competência é requerida?

Q6.4: Em quais posições da organização uma determinada competência é requerida?

Q6.5: A que domínio de conhecimento pertence um conhecimento?

QG7: Quais são os objetivos estabelecidos para a organização?

Q7.1: Quais são os objetivos da organização?

Q7.2: Quais são os objetivos de uma determinada unidade organizacional?

Q7.3: Quais são os objetivos de uma determinada posição?

Q7.4: Como um determinado objetivo é decomposto?

Q7.5: Qual é a prioridade de um determinado objetivo?

QG8: Como têm sido conduzidos os projetos da organização?

Q8.1: Quais são os projetos conduzidos pela organização?

Q8.2: Quais são as outras organizações envolvidas em um determinado projeto?

Q8.3: Qual processo orienta a execução de um determinado projeto?

Q8.4: Qual é a equipe de um determinado projeto?

Para iniciar a captura da ontologia, termos e frases potencialmente relevantes foram identificados e atribuídos a áreas de trabalho, que tiveram sua semântica definida e deram origem a sub-ontologias. A figura 5.6 mostra as sub-ontologias obtidas e os relacionamentos entre elas.

Para cada uma das sub-ontologias, foram elaborados modelos utilizando o sub-conjunto da UML proposto por MIAN [191] para descrição de ontologias. Além disso, os conceitos, relações e restrições foram descritos em linguagem natural e exemplificados. Na formalização, foi utilizada lógica de primeira ordem, definindo-se as constantes, predicados e axiomas. A avaliação da ontologia ocorreu ao longo de todo

o processo. A seguir, são apresentadas a definição e formalização de cada sub-ontologia.

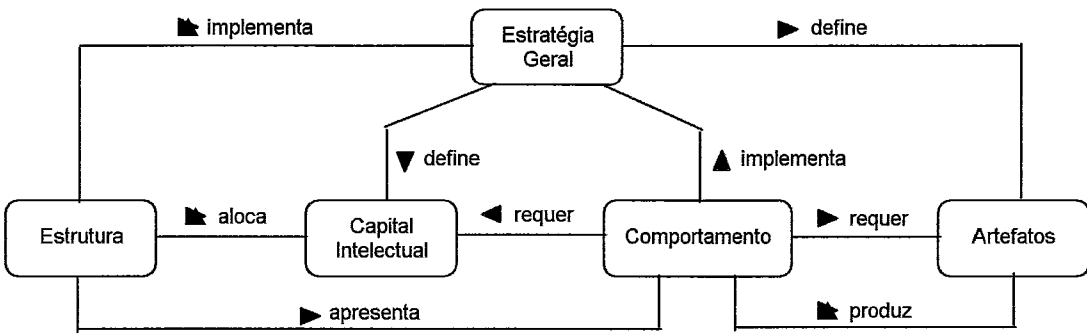


Figura 5.6 – Sub-ontologias da Ontologia de Organização

➤ Sub-ontologia de Capital Intelectual

A sub-ontologia de Capital Intelectual estabelece o vocabulário necessário para descrever o capital intelectual de uma organização, respondendo às questões de competência Q6.1 e Q6.5. Os seguintes aspectos são tratados pela sub-ontologia: taxonomia de competência, interação entre experiência e conhecimento, disponibilidade de competências e decomposição de domínio de conhecimento.

Taxonomia de Competência

Competências tornam as pessoas capazes de executar atividades que envolvem algum grau de dificuldade. A taxonomia de competência (figura 5.7) define como competências podem ser classificadas quanto a sua natureza, de acordo com o propósito da ontologia.

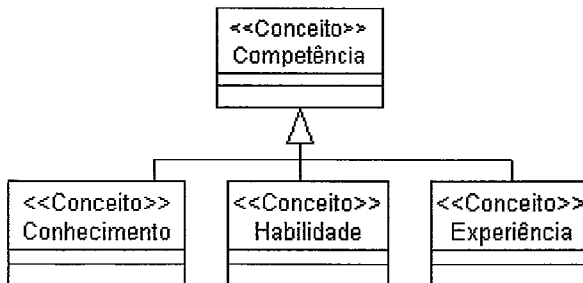


Figura 5.7 – Taxonomia de Competência

Conhecimentos são apropriações de objetos pelo pensamento através de definição, percepção clara, análise, apreensão completa ou outra forma de apropriação [192]. Exemplos de conhecimento são: conhecimento sobre análise orientada a objeto, conhecimento sobre arquiteturas cliente-servidor, conhecimento sobre as teorias de administração e conhecimento sobre as normas de tráfego aéreo.

Habilidades são aptidões natas ou adquiridas não associadas a uma atividade ou domínio de conhecimento específico. Exemplos de habilidades são: liderança, habilidade de negociação, habilidade de argumentação e habilidade de expor idéias em público.

Experiências são adquiridas através da prática, ou seja, através da execução de atividades. Exemplos de experiências são: experiência na definição de arquiteturas cliente-servidor, experiência na administração de aeroportos e experiência na assistência emergencial de pacientes com problemas cardíacos.

Para formalizar a taxonomia de competência, foram definidos os predicados **competência(p)**, **conhecimento(c)**, **habilidade(h)** e **experiência(x)**, que indicam, respectivamente, que **p** é uma competência, **c** é um conhecimento, **h** é uma habilidade e **x** é uma experiência. Além disso, foram definidos os seguintes axiomas:

$(\forall c) (\text{conhecimento}(c) \rightarrow \text{competência}(c))$	(A1)
$(\forall h) (\text{habilidade}(h) \rightarrow \text{competência}(h))$	(A2)
$(\forall x) (\text{experiência}(x) \rightarrow \text{competência}(x))$	(A3)

Axiomas, como estes, que descrevem as restrições impostas pela forma de estruturação dos conceitos, são chamados axiomas epistemológicos e podem ser derivados automaticamente por uma ferramenta de edição de ontologias [123]. Desta forma, sempre que a notação para sub-tipos for utilizada, axiomas deste tipo serão assumidos, não sendo mais apresentados.

Interação entre Experiência e Conhecimento

A figura 5.8 mostra como a interação entre experiência e conhecimento é tratada pela sub-ontologia. Experiências geralmente são relevantes por representarem a aplicação de conhecimentos na prática. Por exemplo, a experiência na definição de arquiteturas cliente-servidor representa a aplicação do conhecimento sobre este tipo de arquitetura na prática. No entanto, existem experiências que são relevantes por outros motivos, tais como a experiência na elaboração de projetos visuais de *sites* para a Internet e a experiência na criação de peças publicitárias, que são relevantes pela criatividade envolvida. Além disso, experiências podem ser descritas de forma tão genérica, que pode não ser de interesse especificar o que a experiência significa em termos de aplicação de conhecimentos. Desta forma, não foi estabelecida nenhuma restrição para a relação proposta.

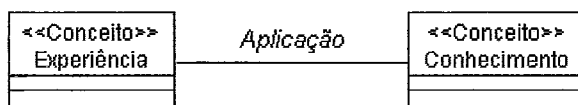


Figura 5.8 – Interação entre Experiência e Conhecimento

Para formalizar a relação, foi definido o predicado *aplicação(x,c)*, indicando que a experiência *x* envolve a aplicação do conhecimento *c*, sendo que o seguinte axioma de consolidação deve ser observado.

$$(\forall c,x) (aplicação(x,c) \rightarrow experiência(x) \wedge conhecimento(c)) \quad (A4)$$

Axiomas de consolidação têm como objetivo verificar a coerência das informações existentes, não representando, neste sentido, conseqüências lógicas, ou seja, não derivando novas informações [123]. Axiomas de consolidação que apenas estabelecem os tipos de objetos que podem ser utilizados como argumentos em um predicado, como o axioma acima, não serão mais apresentados ao longo da descrição desta ontologia.

Disponibilidade de Competências

A figura 5.9 mostra a parte do modelo da sub-ontologia que trata o aspecto da disponibilidade de competências na organização.

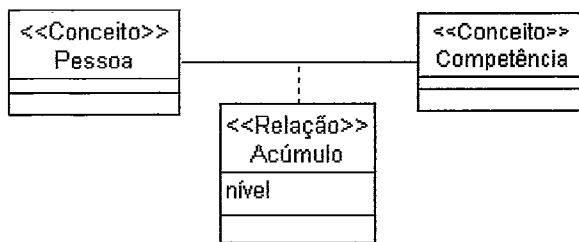


Figura 5.9 – Disponibilidade de Competências

Pessoas são fundamentais para o funcionamento de uma organização, atuando na execução de atividades necessárias ao cumprimento da missão da organização.

Ao longo de sua trajetória profissional, as pessoas acumulam competências, disponibilizando-as para a organização em que trabalham. As competências possuídas pelos profissionais de uma organização são de grande importância para os próprios profissionais, pois são utilizadas para estabelecer o seu papel e o seu valor na organização, e também para a organização, pois representam o seu capital intelectual.

O predicado *peessoa(p)*, que indica que *p* é uma pessoa, e o predicado *acúmulo(p,c,n)*, que indica que a pessoa *p* possui a competência *c* no nível *n*, foram definidos para formalizar o modelo da figura 5.9.

Decomposição de Domínio de Conhecimento

O último aspecto tratado pela sub-ontologia é a decomposição de um domínio de conhecimento (figura 5.10).

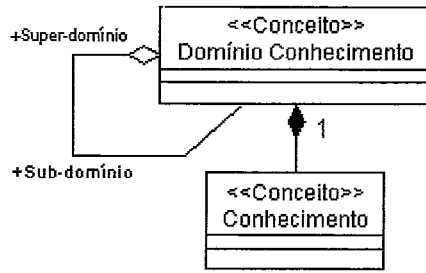


Figura 5.10 – Decomposição de Domínio de Conhecimento

Um **domínio de conhecimento** é um conjunto de conhecimentos reunidos de acordo com a homogeneidade de conteúdo. Exemplos de domínios de conhecimento são: cardiologia, aviação comercial e extração de petróleo.

Para formalizar a relação de composição entre um domínio de conhecimento e os conhecimentos que compõem este domínio, foram definidos o predicado **domínio(*d*)**, que indica que *d* é um domínio de conhecimento, e o predicado **domínio_composição(*c*,*d*)**, que indica que o conhecimento *c* faz parte do domínio de conhecimento *d*. Os axiomas a seguir são um reflexo da restrição expressa pela cardinalidade.

$(\forall c) (\text{conhecimento}(c) \rightarrow (\exists d) (\text{domínio_composição}(c,d)) \quad (\text{A5})$
$(\forall c, d_1, d_2) (\text{domínio_composição}(c, d_1) \wedge \text{domínio_composição}(c, d_2) \rightarrow d_1 = d_2) \quad (\text{A6})$

Além disso, domínios de conhecimentos mais amplos, chamados **super-domínios**, podem ser compostos a partir de outros domínios de conhecimento, chamados **sub-domínios**, incorporando os seus conhecimentos. Por exemplo, o super-domínio de Computação é composto dos sub-domínios de Engenharia de Software, Redes de Computadores, Banco de Dados, entre outros.

Desta forma, foram definidos os predicados **subdomínio(*d*₁,*d*₂)**, que indica que o domínio de conhecimento *d*₁ é um sub-domínio (ou é uma parte) do domínio de conhecimento *d*₂, e **superdomínio(*d*₂,*d*₁)**, que indica que *d*₂ é um super-domínio de (ou é um todo, cuja uma das partes é) *d*₁. Estes predicados estão relacionados segundo a seguinte sentença:

$(\forall d_1, d_2) (\text{subdomínio}(d_1, d_2) \leftrightarrow \text{superdomínio}(d_2, d_1)) \quad (\text{A7})$
--

A notação para todo-parte, quando utilizada para representar uma relação todo-parte entre instâncias do mesmo conceito, implica na existência dos axiomas abaixo:

$$(\forall d_1, d_2, d_3) (\text{subdomínio}(d_1, d_2) \wedge \text{subdomínio}(d_2, d_3) \rightarrow \text{subdomínio}(d_1, d_3)) \quad (\text{A8})$$

$$(\forall d_1, d_2) (\text{subdomínio}(d_1, d_2) \rightarrow \neg \text{subdomínio}(d_2, d_1)) \quad (\text{A9})$$

O primeiro axioma (axioma A8) formaliza a propriedade transitiva. Segundo esta propriedade, se d_1 é um sub-domínio do domínio de conhecimento d_2 e d_2 é um sub-domínio do domínio de conhecimento d_3 , então d_1 é também um sub-domínio de d_3 . Além disso, a relação todo-parte é assimétrica, o que é formalizado pelo segundo axioma (axioma A9), ou seja, se um domínio de conhecimento d_1 é um sub-domínio do domínio de conhecimento d_2 , então d_2 não pode ser um sub-domínio de d_1 .

Axiomas epistemológicos como os axiomas de A5 a A9 não serão mais apresentados no decorrer deste trabalho, pois podem ser derivados por uma ferramenta para edição de ontologias [123].

Os predicados **domínioelementar(d)** e **macrodomínio(d)** indicam, respectivamente, que d é um domínio que não pode ser decomposto em outros domínios de conhecimento, e que d é um domínio que não faz parte de nenhum outro domínio de conhecimento. Estes predicados são definidos em termos dos conceitos de sub-domínio e super-domínio, como mostram as sentenças abaixo.

$$(\forall d) (\text{domínioelementar}(d) \leftrightarrow \neg (\exists d_1) (\text{subdomínio}(d_1, d))) \quad (\text{A10})$$

$$(\forall d) (\text{macrodomínio}(d) \leftrightarrow \neg (\exists d_1) (\text{superdomínio}(d_1, d))) \quad (\text{A11})$$

Para garantir a integridade do modelo, de forma que um domínio de conhecimento seja composto ou de um conjunto de sub-domínios ou de um conjunto de conhecimentos, foi definido o axioma de consolidação abaixo, indicando que apenas domínios elementares são decompostos em termos de conhecimentos.

$$(\forall c, d) (\text{domínio_composição}(c, d) \rightarrow \text{domínioelementar}(d)) \quad (\text{A12})$$

Para representar que o conteúdo de um domínio de conhecimento é dado pelo conteúdo de seu sub-domínios (se é um super-domínio) ou pelo conjunto de conhecimentos que o compõem (se é um domínio elementar), foi definido o predicado **domínio_conteúdo(c, d)**, que indica que o conhecimento c faz parte do conteúdo definido pelo domínio de conhecimento d . Para isto, foi definido o seguinte axioma:

$$(\forall c, d) (\text{domínio_conteúdo}(c, d) \leftrightarrow \text{domínio_composição}(c, d) \vee (\exists d_k) (\text{subdomínio}(d_k, d) \wedge \text{domínio_conteúdo}(c, d_k))) \quad (\text{A13})$$

➤ Sub-ontologia de Estrutura

A sub-ontologia de Estrutura estabelece o vocabulário necessário para descrever como a organização está estruturada, respondendo às questões de competência Q2.1 a Q2.4, Q3.1 a Q3.3, Q5.1 a Q5.5, Q6.3, Q6.4 e Q7.1 a Q7.5. Com base nestas questões de competência, foi verificada a necessidade de se tratar os seguintes aspectos: decomposição da organização, distribuição de autoridade e responsabilidade entre unidades organizacionais, decomposição de unidade organizacional, distribuição de autoridade e responsabilidade entre posições, especificação de cargo e posição, preenchimento de vagas, formação de equipe e definição de objetivos.

Decomposição da Organização

Organizações são grupos de pessoas trabalhando em conjunto para o cumprimento de uma missão, onde a missão de uma organização define o seu propósito dentro do sistema econômico ou social. Segundo MEGGINSON *et al.* [24], a divisão do trabalho é o princípio base das organizações, permitindo que grupos de pessoas, trabalhando em conjunto de modo cooperativo e coordenado, possam realizar mais do que cada uma agindo de forma independente. Existem diversas maneiras de estruturar uma organização de forma a estabelecer a divisão de trabalho [24], mas os componentes básicos são os cargos, as unidades organizacionais e as comissões. A figura 5.11 aborda o aspecto da decomposição da organização.

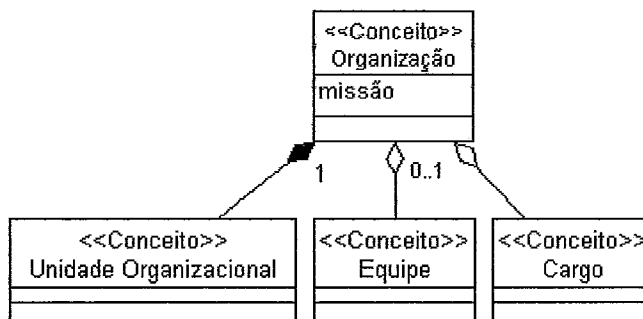


Figura 5.11 – Decomposição da Organização

Unidades Organizacionais são agrupamentos de componentes da organização (por exemplo: atividades e pessoas) de acordo com a homogeneidade de conteúdo¹¹ para que a organização possa ser econômica e eficiente (baseado em [188]).

¹¹ A Teoria Geral de Administração prevê agrupamento por funções (por exemplo: produção, contabilidade, vendas e serviço ao cliente), por produtos ou serviços, por localização geográfica, por clientela, por processos, por projetos, ressaltando que estes tipos de agrupamento podem ser insuficientes para resolver todos os casos e que a maioria das organizações utiliza mais de um tipo de agrupamento [24,188].

Exemplos de unidades organizacionais são: divisão farmacêutica, departamento de antibióticos, agência em Brasília e seção de contas corrente.

Equipes são agrupamentos de pessoas com finalidade determinada e, normalmente, por período de tempo limitado. O conceito de Equipe, no contexto desta sub-ontologia, tem o objetivo de possibilitar a representação de comissões. Segundo CHIAVENATO [187], comissões se distinguem das unidades organizacionais por geralmente terem finalidade que abrange várias unidades organizacionais (assuntos interdepartamentais), terem participantes que pertencem a diferentes unidades organizacionais e diferentes níveis hierárquicos, não terem posição definida na estrutura organizacional, funcionarem esporádica ou intermitentemente durante certos dias ou em determinadas horas, e durarem enquanto não atingirem seu objetivo ou não realizarem a tarefa para a qual foram criadas. Como será visto na sub-ontologia de Comportamento, o conceito de Equipe também pode estar relacionado a projetos da organização. Sendo assim, o conceito de **Comissão** foi definido como sendo uma equipe que compõe a organização sem estar relacionada a um projeto. Segundo MEGGINSON *et al.* [24], alguns exemplos de comissões são comissões orçamentárias, conselhos administrativos e comissão de planejamento de um novo produto.

O axioma de consolidação que reflete a restrição implícita na definição de comissão será apresentado posteriormente (axioma A51), pois depende de predicados que extrapolam o escopo desta sub-ontologia.

Cargos especificam o conjunto de atividades a serem executadas pelas pessoas que os ocupam, suas responsabilidades, competências desejadas, além das condições de trabalho oferecidas, mas não determinam as unidades organizacionais específicas às quais essas pessoas serão alocadas. Não foi adotado o termo Função para este conceito, pois este termo tem um outro significado no contexto da Administração, conforme pode ser observado na nota da página anterior.

Distribuição de Autoridade e Responsabilidade entre Unidades Organizacionais

A figura 5.12 mostra como a distribuição de autoridade e responsabilidade entre unidades organizacionais é tratada na sub-ontologia.

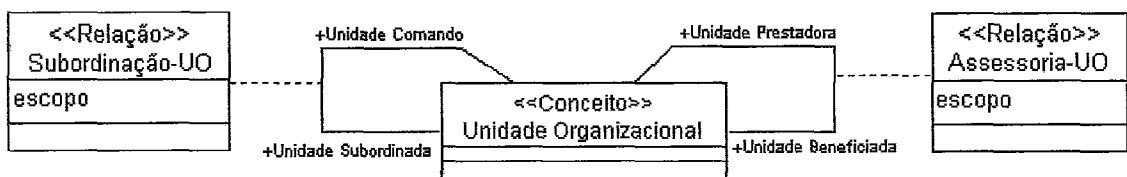


Figura 5.12 – Distribuição de Autoridade e Responsabilidade

O modelo proposto permite: 1) a autoridade linear ou única, em que cada unidade subordinada reporta-se única e exclusivamente a uma única unidade superior; 2) a autoridade funcional ou dividida, em que cada unidade subordinada reporta-se a várias unidades superiores, porém reporta-se a cada uma delas somente nos assuntos da especialidade de cada uma (ou seja, dentro de um escopo); e 3) a fusão das abordagens anteriores com predomínio da primeira, que é chamada linha-*staff*, em que cada unidade organizacional reporta-se a uma única unidade superior (linha), porém recebe assessoria e serviços especializados de várias unidades organizacionais (*staff*), que possuem autoridade relativa e parcial estabelecida mediante uma relação funcional e de consultoria [187]. A figura 5.13 fornece uma representação esquemática dessas relações.

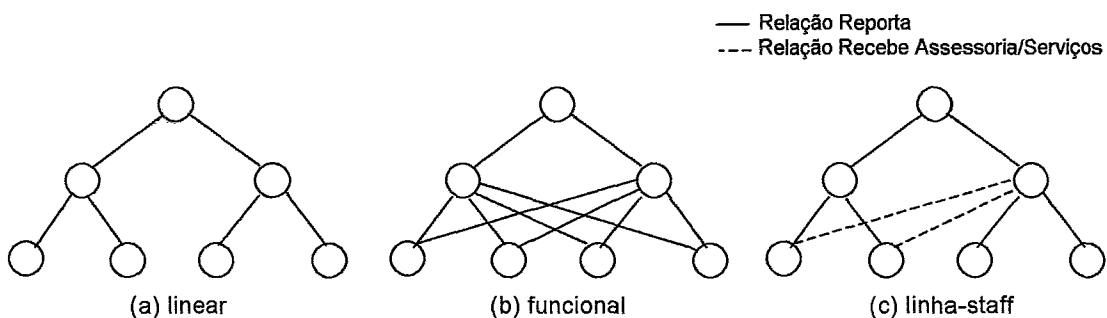


Figura 5.13 – Esquema das Estruturas adotadas nas Organizações

Um exemplo de unidade organizacional que tem relação de assessoria e prestação de serviços com outras unidades em organizações que desenvolvem e mantêm software é a Gerência da Qualidade, cuja atividade não é decidir, comandar, executar, mas sim pensar, planejar, sugerir, recomendar, assessorar e prestar serviços especializados relacionados à qualidade.

Para formalizar as relações representadas na figura 5.12, foram definidos o predicado **subordinação_UO**(u_1, u_2, s), que indica que a unidade organizacional u_1 se reporta à unidade organizacional u_2 dentro do escopo definido por s , e o predicado **assessoria_UO**(u_1, u_2, s), que indica que a unidade organizacional u_1 assessoria ou presta serviço à unidade organizacional u_2 dentro do escopo definido por s . Também foram definidos os predicados **UOsubordinada**(u_1, u_2), **UOcomando**(u_2, u_1), **UObeneficiada**(u_1, u_2), **UOprestadora**(u_2, u_1), indicando os papéis representados pelas unidades nas duas relações acima. Estes predicados obedecem a axiomas similares ao axioma A9, pois tanto a relação de subordinação quanto a relação de assessoria são assimétricas. Além disso, a relação de subordinação também é transitiva, de modo que foi definido um axioma similar ao A8 para o predicado

$UO_{subordinada}(u_1, u_2)$, que, conseqüentemente, também é válido para o predicado $UO_{comando}(u_2, u_1)$.

Decomposição de Unidade Organizacional

A figura 5.14 apresenta o modelo referente à decomposição de unidade organizacional.

Posições são o resultado não só da alocação de cargos a unidades organizacionais, mas, principalmente, da definição da estrutura necessária para que a unidade organizacional funcione de forma eficiente e econômica, especificando atividades, responsabilidades e competências adicionais, em conformidade com a finalidade da unidade organizacional específica. Em uma universidade federal, por exemplo, departamentos são unidades organizacionais e professor assistente, professor adjunto e professor titular são alguns dos cargos existentes. Ao serem alocados a um departamento (por concurso ou por empréstimo entre departamentos), os professores passam a ocupar posições no departamento. Uma das posições, por exemplo, é a de chefe de departamento, que tem como propósito coordenar as ações dos professores e demais funcionários alocados ao departamento. Chefe de departamento em uma universidade não é um cargo, mas sim uma posição que é ocupada por um professor por um período de tempo.

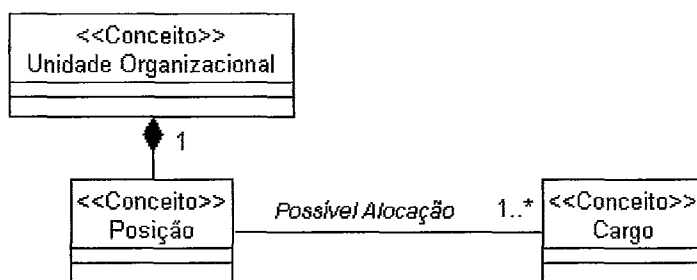


Figura 5.14 – Decomposição de Unidade Organizacional

O predicado **possível alocação(p,c)** indica que a posição p representa a possibilidade de alocação de pessoas com o cargo c na unidade à qual p pertence, ou seja, que a posição p pode ser ocupada por pessoas que possuem o cargo c .

Distribuição de Autoridade e Responsabilidade entre Posições

O aspecto da distribuição de autoridade e responsabilidade entre posições é capturado pelas relações descritas no modelo da figura 5.15, que contemplam as mesmas estruturas discutidas no nível de unidades organizacionais, pois estas estruturas são aplicáveis em ambos os níveis, conforme discutido por CHIAVENATO [187]. Isto implica que predicados e axiomas similares aos definidos para formalizar as relações da figura 5.12 foram definidos.

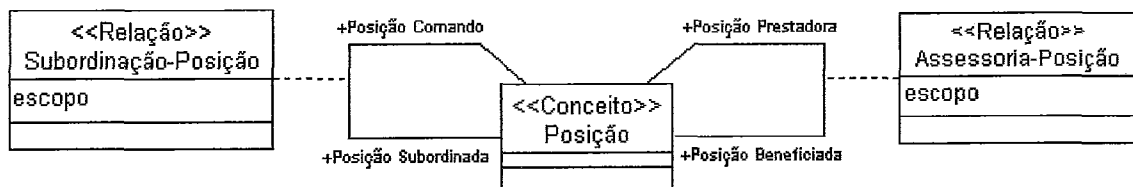


Figura 5.15 – Distribuição de Autoridade e Responsabilidade

Especificação de Cargos e Posições

A figura 5.16 ilustra como a especificação de cargos e posições é abordada na sub-ontologia.

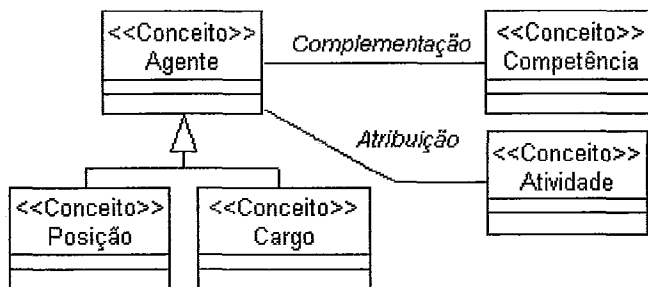


Figura 5.16 – Especificação de Cargos e Posições

Tanto cargos quanto posições especificam competências desejadas, atividades a serem executadas e responsabilidades a serem assumidas pelas pessoas que fazem parte da organização. Desta forma, foi introduzido o conceito de **Agente** na sub-ontologia, que é uma especificação de um perfil necessário para que a organização cumpra a sua missão através da execução de atividades. Um agente pode representar um cargo ou uma posição.

Os predicados *atribuição(a,g)* e *complementação(c,g)* foram definidos, para indicar, respectivamente, que *a* é uma atividade a ser executada pelo agente *g* (ou seja, que lhe foi atribuída e lhe caracteriza), e que o perfil definido pelo agente *g* especifica que a competência *c* é desejada, em complementação às competências requeridas pelas atividades que lhe foram atribuídas. Os axiomas necessários para derivar as competências desejadas em um agente serão apresentados posteriormente (axioma A31), pois dependem de predicados que extrapolam o escopo desta sub-ontologia.

Preenchimento de Vagas e Formação de Equipes

O preenchimento das vagas em uma organização é representado pela alocação de pessoas às posições existentes e a conseqüente ocupação dos possíveis cargos, considerando, para isso, as competências possuídas pelas pessoas e as competências requeridas pelas posições e cargos. A figura 5.17 aborda este aspecto da sub-ontologia e o aspecto da formação de equipe.

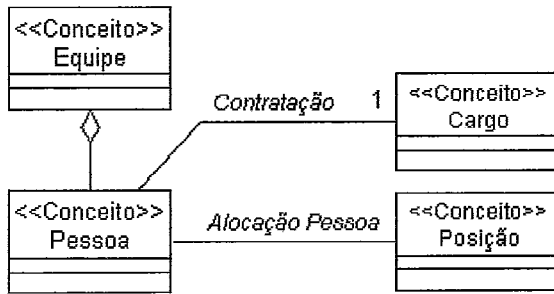


Figura 5.17 – Preenchimento de Vagas e Formação de Equipe

Para formalização das relações representadas na figura 5.17, foram definidos os predicados *contratação(p,c)* e *alocação_pessoa(p,s)*, que indicam, respectivamente, que a pessoa *p* foi contratada para exercer o cargo *c* e que a pessoa *p* foi alocada à posição *s*, cabendo ressaltar que a relação de contratação representa qualquer forma de ingresso de uma pessoa em uma organização, incluindo contratos temporários e de sociedade, e que pessoas podem não estar alocadas a uma posição, uma vez que micro e pequenas empresas podem não ser decompostas em unidades organizacionais.

Uma observação a ser feita é que não foram definidos axiomas para restringir o preenchimento de vagas em uma organização com relação ao cargo necessário para ocupar uma posição ou as competências necessárias para ocupar um cargo ou posição, pois estas restrições nem sempre são obedecidas nas organizações.

Definição de Objetivos

A definição de objetivos é o último aspecto tratado por esta sub-ontologia (figura 5.18).

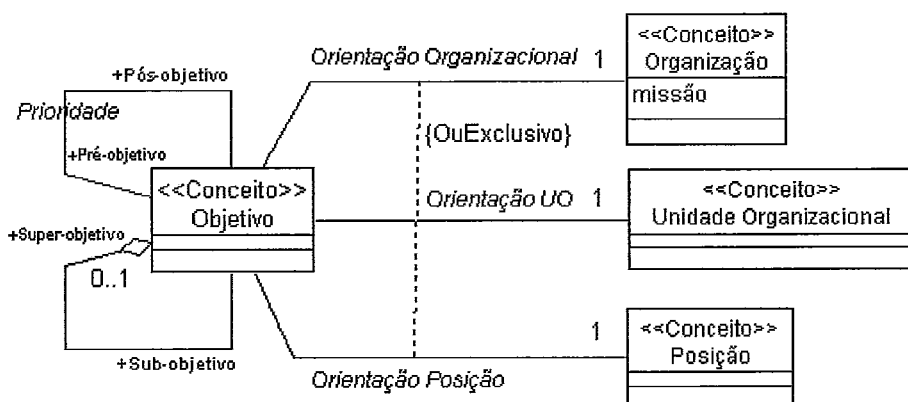


Figura 5.18 – Definição de Objetivos

Objetivos são enunciados escritos sobre os resultados a serem alcançados num período de tempo determinado, onde os resultados devem ser, o quanto possível, quantificáveis [189].

Apesar de alguns autores preferirem diferenciar o conceito de Objetivo do conceito de Meta, MEGGINSON *et al.* [190] defende que é muito mais comum não diferenciá-los. Tanto as metas como os objetivos fornecem orientação e refletem as condições desejadas para melhoria do desempenho global da organização. Enquanto os objetivos são mais amplos, as metas são mais específicas, sendo mais adequadas para orientar as tomadas de decisão e atividades cotidianas da organização. A relação de composição de objetivos estabelece que um objetivo, chamado **super-objetivo**, pode ser decomposto em outros objetivos, chamados **sub-objetivos**, o que permite que os objetivos iniciais, estabelecidos em termos gerais, possam ser detalhados até estarem adequados para orientar as tomadas de decisão e atividades cotidianas da organização. Sendo assim, não foi considerado necessário diferenciar estes conceitos no contexto desta ontologia.

Objetivos podem ser estabelecidos para a organização como um todo, para suas unidades organizacionais, e uma unidade organizacional pode, ainda, estabelecer objetivos a serem alcançados pelas suas posições, de forma que os objetivos da unidade organizacional sejam alcançados [189,190]. Objetivos organizacionais podem estar relacionados à posição competitiva no mercado, índices de eficiência, taxa de dividendos, inovação e criatividade nos produtos, responsabilidade pública e social, entre outros. Exemplos de objetivos organizacionais são: “Administrar a empresa de modo que ela permaneça como líder mundial no campo de abrasivos”, “Manter e construir a reputação da empresa como empresa responsável”.

A maioria das organizações possui múltiplos objetivos. Unidades organizacionais e posições também podem possuir múltiplos objetivos. A existência de múltiplos objetivos pode acarretar conflitos entre os mesmos, sendo necessário, portanto, estabelecer o equilíbrio ótimo entre eles [189,190]. Além da redefinição dos objetivos, uma forma de equilibrá-los é estabelecer prioridades entre eles. Segundo CHIAVENATO [189], objetivos precisam ser escalonados em ordem gradativa de prioridade. A relação de prioridade entre objetivos estabelece que um objetivo, chamado **pré-objetivo**, é mais importante que um outro objetivo, chamado **pós-objetivo**, o que permite a solução dos possíveis conflitos.

Para formalizar as relações mencionadas, foram definidos: 1) o predicado **subobjetivo(t_1, t_2)**, que indica que o objetivo t_1 é um sub-objetivo do (ou é um meio de atingir o) objetivo t_2 , e o correspondente **superobjetivo(t_2, t_1)**; 2) o predicado **preobjetivo(t_1, t_2)**, que indica que t_1 é um pré-objetivo de (ou deve ser priorizado em relação a) t_2 , e o correspondente **posobjetivo(t_2, t_1)**; e 3) os predicados **orientação_org(t, r)**, **orientação_UO(t, u)** e **orientação_pos(t, p)**, que indicam,

respectivamente, que o objetivo t orienta a execução das atividades da organização r , da unidade u e da posição p .

A notação utilizada na figura 5.18 para expressar o condicionante Ou-Exclusivo também permite que uma ferramenta de edição de ontologias derive os axiomas epistemológicos pertinentes, que são:

$(\forall t, r) (orientação_org(t, r) \rightarrow (\neg \exists u) orientação_UO(t, u) \wedge$	(A14)
$(\neg \exists p) orientação_pos(t, p))$	
$(\forall t, u) (orientação_UO(t, u) \rightarrow (\neg \exists r) orientação_org(t, r) \wedge$	(A15)
$(\neg \exists p) orientação_pos(t, p))$	
$(\forall t, p) (orientação_pos(t, p) \rightarrow (\neg \exists r) orientação_org(t, r) \wedge$	(A16)
$(\neg \exists u) orientação_UO(t, u))$	

No entanto, devem ser, ainda, observadas as restrições que estabelecem que a relação de prioridade sempre envolve objetivos do mesmo nível (axiomas A18 e A19), sendo que, no nível de organização (axioma A17), a restrição é ainda mais forte e os objetivos relacionados devem pertencer a uma mesma organização, pois objetivos de organizações diferentes são completamente independentes entre si. Já as restrições sobre a relação de composição (axiomas A20 a A22) estabelecem que esta relação sempre envolve objetivos que orientam as atividades do mesmo elemento da estrutura organizacional, quer este seja a organização, uma unidade organizacional ou uma posição.

$(\forall t_1, t_2, r) (orientação_org(t_1, r) \wedge preobjetivo(t_1, t_2) \rightarrow$	(A17)
$orientação_org(t_2, r))$	
$(\forall t_1, t_2, u_1) (orientação_UO(t_1, u_1) \wedge preobjetivo(t_1, t_2) \rightarrow$	(A18)
$(\exists u_2) (orientação_UO(t_2, u_2)))$	
$(\forall t_1, t_2, p_1) (orientação_pos(t_1, p_1) \wedge preobjetivo(t_1, t_2) \rightarrow$	(A19)
$(\exists p_2) (orientação_pos(t_2, p_2)))$	
$(\forall t_1, t_2, r) (orientação_org(t_1, r) \wedge subobjetivo(t_1, t_2) \rightarrow$	(A20)
$orientação_org(t_2, r))$	
$(\forall t_1, t_2, u) (orientação_UO(t_1, u) \wedge subobjetivo(t_1, t_2) \rightarrow$	(A21)
$orientação_UO(t_2, u))$	
$(\forall t_1, t_2, p) (orientação_pos(t_1, p) \wedge subobjetivo(t_1, t_2) \rightarrow$	(A22)
$orientação_pos(t_2, p))$	

De forma similar à composição de domínio de conhecimento, foram definidos os predicados **objetivoelementar(t)** e **macroobjetivo(t)**.

➤ Sub-ontologia de Artefatos

Artefato é qualquer elemento produzido pelo homem e não por causas naturais, podendo exercer diferentes papéis em uma organização, tais como o de insumo ou produto de uma atividade. A sub-ontologia de Artefatos agrupa os conceitos e relações que definem os artefatos em termos de sua natureza e composição. Considerada de forma isolada, a sub-ontologia não responde a nenhuma questão de competência, mas é fundamental para que as questões Q1.5, Q4.4 e Q4.5 sejam respondidas.

Decomposição de Artefato

A figura 5.19 ilustra a relação de composição entre artefatos, que estabelece que um artefato, chamado **super-artefato**, pode ser decomposto em outros artefatos, chamados **sub-artefatos**.

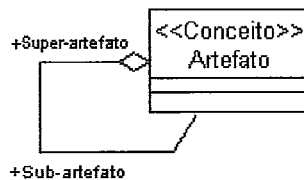


Figura 5.19 – Decomposição de Artefato

De forma similar à composição de domínio de conhecimento, foram definidos os predicados **subartefato(s_1, s_2)**, **superartefato(s_2, s_1)**, **artefatoelementar(s)** e **macroartefato(s)**.

Taxonomia de Artefato

A taxonomia de artefato (figura 5.20) define como um artefato pode ser classificado quanto a sua natureza, de acordo com o propósito da ontologia.

Documentos são artefatos escritos ou que podem ser impressos, cuja função é fornecer informação, conhecimento ou prova¹². Exemplos de documentos são especificações de requisitos, a norma ISO/IEC 12207, documentos de identificação pessoal, ordens de serviço e de pagamento.

Bens são artefatos concluídos, no sentido de não participarem da composição de outros artefatos, além de não serem documentos. Em outras palavras, bens são artefatos que já passaram por todas as transformações necessárias para que incorporassem suas funcionalidades e que, conseqüentemente, fazem parte do seu processo de produção. Bens podem ser classificados em bens de usufruto e bens de produção.

¹² Definição estabelecida com base na definição fornecida em [193].

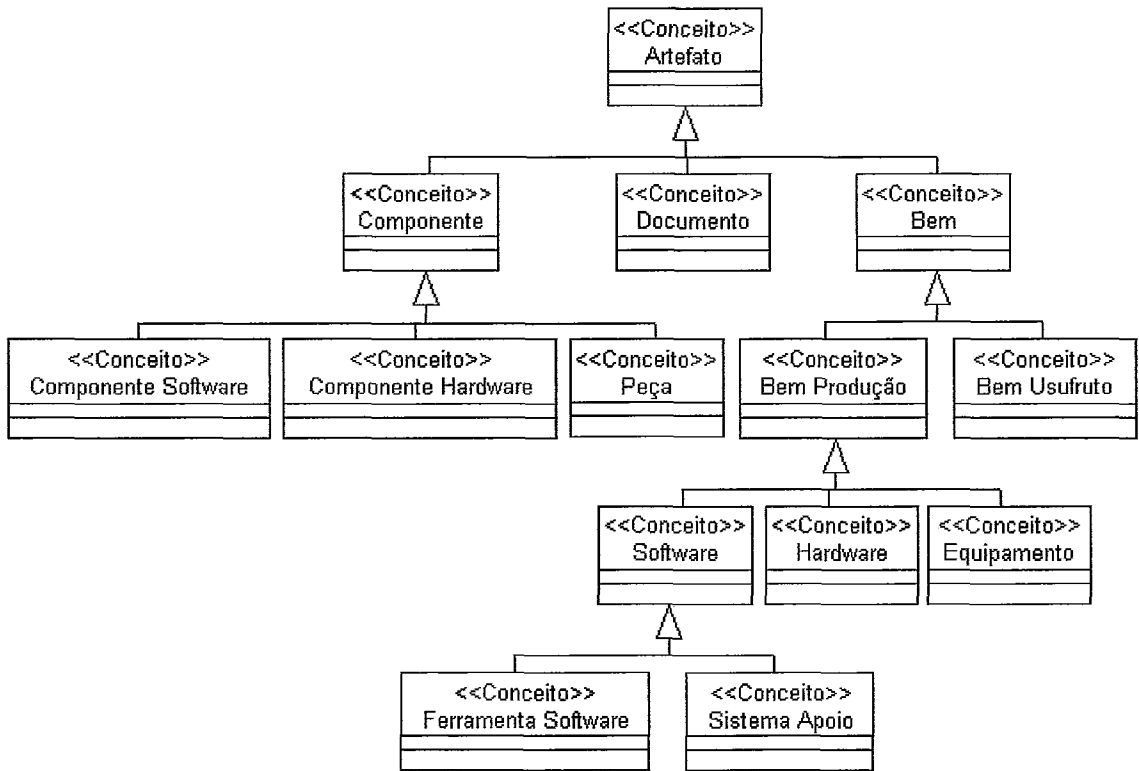


Figura 5.20 – Taxonomia de Artefato

Bens de Usufruto são bens dos quais se pode usufruir as funcionalidades, que não são voltadas para a criação ou transformação de artefatos. Exemplos de bens de usufruto são jogos eletrônicos, aparelhos de som e carros.

Bens de Produção são bens cujas funcionalidades apóiam a criação ou a transformação de artefatos. Bens de produção podem ser classificados em software, hardware e equipamentos.

Software é um bem de produção representado por um conjunto de instruções e dados que, utilizado em conjunto com um hardware, é capaz de executar ou apoiar a execução de atividades. Artefatos de Software¹³ podem ser classificados em ferramentas de software e sistemas de apoio, que são definidos conforme proposto por FALBO [123].

Ferramentas de Software são artefatos de software utilizados para (semi-) automatizar procedimentos. Exemplos de ferramentas de software que (semi-) automatizam procedimentos de desenvolvimento e manutenção de software são ferramentas de programação, sistemas gerenciadores de bancos de dados, ferramentas para coleta automática de métricas, geradores de casos de teste e sistemas de controle de versões. Para organizações com outro tipo de negócio, os

¹³ O termo “Artefatos de software” é utilizado como plural do termo “Software”, assim como o termo “Artefatos de hardware” será utilizado como plural do termo “Hardware”.

exemplos de ferramentas de software são sistemas de controle de caldeiras, ferramentas de CAD (*Design Assisted by Computer*) e sistemas de conta corrente. Além disso, existem as ferramentas de software de propósito geral, tais como editores de texto, editores de formulário e editores de figuras.

Sistemas de Apoio são artefatos de software que não (semi-)automatizam procedimentos, mas, por serem necessários à operação adequada do hardware ou de outros artefatos de software, são requeridos para a execução de atividades da organização. Exemplos de sistemas de apoio são sistemas operacionais e sistemas de gerenciamento de redes.

De volta aos tipos de bens de produção, **hardware** é um bem de produção representado por um computador, um de seus periféricos ou uma máquina qualquer operada com o auxílio de um software, e **equipamento** é um bem de produção, sem ser um software ou hardware, requerido para a execução de uma atividade, podendo ser uma máquina ou instrumento qualquer.

Componentes são artefatos que são utilizados na composição de outros artefatos, mas que não são documentos. Componentes podem ser classificados em componentes de software, componentes de hardware e peças.

Componentes de Software são constituídos por conjuntos de instruções e dados, podendo ser produzidos em uma atividade de desenvolvimento de software ou ser obtidos de uma biblioteca de componentes. Exemplos de componentes de software são funções, sub-programas, classes, *frameworks*, padrões gerativos.

Componentes de Hardware são componentes especificamente utilizados na composição de artefatos de hardware. Exemplos são: unidades de entrada e saída, circuitos integrados e discos.

Peças são componentes de propósito geral, que podem entrar na composição de equipamentos ou de artefatos de hardware. Exemplos são: parafusos, correias, manivelas e motores.

Para formalizar a taxonomia de artefato, foram definidos os seguintes predicados: **componente(s,t)**, **documento(s)**, **bem(s)**, **bemprodução(s)**, **bemusufruto(s)**, **software(s)**, **hardware(s)**, **equipamento(s)**, **ferramentasoft(s)**, **sistemaapoio(s)**, sendo que **componente(s,t)** indica que **s** é um componente do tipo **t**, onde **t** assume um dos seguintes valores {*CompSoftware, CompHardware, Peça*}. Os demais foram definidos de forma similar à definição de predicados para a taxonomia de competência. Além disso, foram formalizados os seguintes axiomas:

- Bens são, por definição, macro-artefatos (axioma A23);
- Um componente só pode ser composto por outros componentes do mesmo tipo (axioma A24);

- Um documento só pode ser composto por outros documentos (axioma A25);
- Um software só pode ser composto de componentes de software ou de documentos (axioma A26);
- Um equipamento só pode ser composto de peças e documentos (axioma A27);
- Se um artefato é software e tem sub-artefatos, então um dos sub-artefatos é um componente de software (axioma A28);
- Se um artefato é hardware e tem sub-artefatos, então um dos sub-artefatos é um componente de hardware (axioma A29);
- Se um artefato é equipamento e tem sub-artefatos, então um dos sub-artefatos é uma peça (axioma A30).

$$(\forall s) \quad (bem(s) \rightarrow macroartefato(s)) \quad (A23)$$

$$(\forall s_1, s_2, t_1, t_2) \quad (componente(s_1, t_1) \wedge componente(s_2, t_2) \wedge subartefato(s_1, s_2) \rightarrow t_1 = t_2) \quad (A24)$$

$$(\forall s_1, s_2) \quad (documento(s_2) \wedge subartefato(s_1, s_2) \rightarrow documento(s_1)) \quad (A25)$$

$$(\forall s_1, s_2) \quad (software(s_2) \wedge subartefato(s_1, s_2) \rightarrow componente(s_1, CompSoftware) \vee documento(s_1)) \quad (A26)$$

$$(\forall s_1, s_2) \quad (equipamento(s_2) \wedge subartefato(s_1, s_2) \rightarrow componente(s_1, Peça) \vee documento(s_1)) \quad (A27)$$

$$(\forall s) \quad (software(s) \wedge (\exists s_1, \dots, s_n) (subartefato(s_1, s) \wedge \dots \wedge subartefato(s_n, s)) \rightarrow (\exists s_k) (subartefato(s_k, s) \wedge componente(s_k, CompSoftware))) \quad (A28)$$

$$(\forall s) \quad (hardware(s) \wedge (\exists s_1, \dots, s_n) (subartefato(s_1, s) \wedge \dots \wedge subartefato(s_n, s)) \rightarrow (\exists s_k) (subartefato(s_k, s) \wedge componente(s_k, CompHardware))) \quad (A29)$$

$$(\forall s) \quad (equipamento(s) \wedge (\exists s_1, \dots, s_n) (subartefato(s_1, s) \wedge \dots \wedge subartefato(s_n, s)) \rightarrow (\exists s_k) (subartefato(s_k, s) \wedge componente(s_k, Peça))) \quad (A30)$$

➤ Sub-ontologia de Comportamento

A sub-ontologia de Comportamento responde às questões de competência Q4.1 a Q4.8, Q6.2 e Q8.1 a Q8.4, estando fortemente baseada na ontologia definida por FALBO [123], sendo que os conceitos foram, em geral, definidos de forma mais genérica para poderem ser utilizados para descrever o comportamento de diferentes organizações. Os aspectos abordados pela sub-ontologia são: atividade como ação de transformação, taxonomia de atividade, decomposição de processo e de atividade,

adoção de procedimento, taxonomia de procedimento, método como procedimento sistemático, automatização de procedimento, processos definidos na organização e normas associadas, além de projetos da organização.

Atividade como Ação de Transformação

Uma **atividade** é uma ação de transformação que pode requerer competências para a sua execução e, ao ser executada, fazer uso de bens de produção (recursos), consumir matérias-primas e artefatos de entrada (insumos), além de produzir artefatos de saída (produtos). A figura 5.21 ilustra a parte da sub-ontologia que captura o aspecto da atividade como ação de transformação.

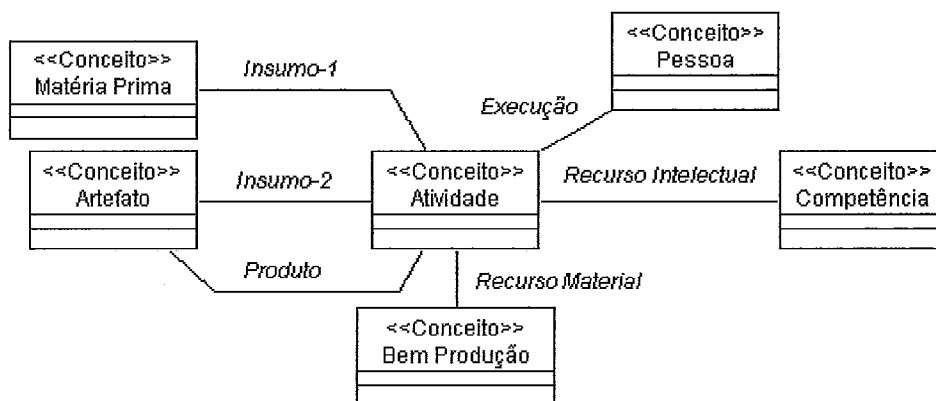


Figura 5.21 – Atividade como Ação de Transformação

Uma **matéria-prima** é uma matéria bruta ou pouco elaborada utilizada como insumo em uma atividade, no sentido de ser um objeto de transformação desta atividade¹⁴.

Para formalizar as relações ilustradas na figura 5.21 foram definidos os predicados ***insumo-1(m,t)***, que indica que a matéria-prima ***m*** é um insumo para a atividade ***t***, ***insumo-2(s,t)***, que indica que o artefato ***s*** é um insumo para a atividade ***t***, ***produto(s,t)***, que indica que o artefato ***s*** é um produto da atividade ***t***, ***recursomaterial(b,t)***, que indica que o bem de produção ***b*** é um recurso material para a atividade ***t***, ***recursointelectual(c,t)***, que indica que a competência ***c*** é um recurso intelectual para a atividade ***t***, além de ***execução(p,t)***, que indica que a pessoa ***p*** é a responsável pela execução da atividade ***t***.

Nenhuma restrição foi estabelecida para a relação de execução de atividades por pessoas, pois nem sempre é desejado estabelecer quem exatamente é responsável pela execução de uma atividade.

Uma vez definida a relação de competência requerida por uma atividade (***recurso intelectual***), foi possível especificar as competências de um agente em

¹⁴ Definição estabelecida com base na definição fornecida em [193].

termos das competências requeridas nas atividades que são de sua responsabilidade, o que é formalizado através do seguinte axioma:

$$(\forall c,g) (competência_agente(c,g) \leftrightarrow (\exists t) (atribuição(t,g) \wedge recursointelectual(c,t) \vee complementação(c,g))) \quad (A31)$$

Além disso, os axiomas abaixo foram formalizados para representar que uma posição, quando relacionada a apenas um cargo através da relação **possívelalocação**, herda as atividades e as competências requeridas para o cargo, podendo acrescentar competências e atividades específicas.

$$(\forall c,p) ((\exists r) (possívelalocação(p,r) \wedge competência_agente(c,r)) \rightarrow competência_agente(c,p)) \quad (A32)$$

$$(\forall p,t) ((\exists r) (possívelalocação(p,r) \wedge atribuição(t,r)) \rightarrow atribuição(t,p)) \quad (A33)$$

Taxonomia de Atividade

A taxonomia de atividade define, de acordo com o propósito da ontologia, como uma atividade pode ser classificada quanto ao papel que desempenha em relação à missão da organização (figura 5.22) e quanto a sua natureza (figura 5.23).

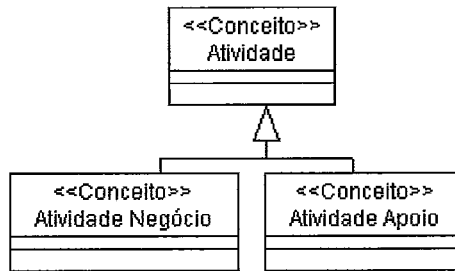


Figura 5.22 – Taxonomia de Atividade quanto ao Papel

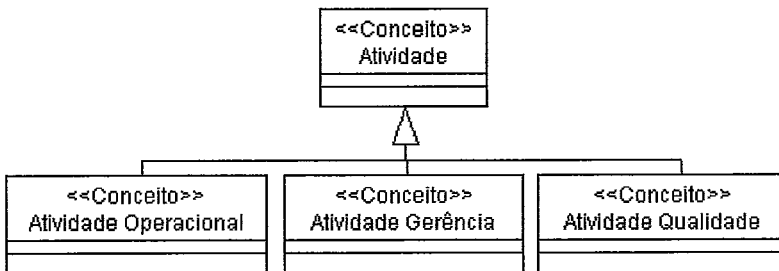


Figura 5.23 – Taxonomia de Atividade quanto à Natureza

Atividades de negócio são atividades essenciais para que a organização cumpra a sua missão e ofereça os artefatos ou serviços a que se propõe. Estas atividades são também chamadas de atividades-fins ou atividades de linha [187].

Atividades de apoio são atividades que buscam oferecer melhores condições para a execução das atividades de negócio. Estas atividades são também chamadas de atividades-meio ou atividades de *staff* [187].

Atividades de gerência são atividades de negócio ou de apoio relacionadas ao planejamento, organização, coordenação e acompanhamento gerencial de outras atividades.

Atividade de avaliação da qualidade são atividades de apoio relacionadas com a garantia da qualidade dos artefatos ou serviços oferecidos pela organização ou dos processos utilizados para oferecer estes artefatos ou serviços.

Atividades operacionais são atividades de negócio ou de apoio responsáveis pelo funcionamento da organização, não sendo, portanto, atividades relacionadas com a garantia da qualidade ou com gerência.

De forma similar à adotada na definição de predicados para a taxonomia de competência, foram definidos os predicados *atvnegócio(t)*, *atvapoio(t)*, *atvgerência(t)*, *atvqualidade(t)* e *atvoperacional(t)* para formalizar os tipos de atividade. O seguinte axioma de consolidação reflete o fato de uma atividade de avaliação da qualidade ser, por definição, uma atividade de apoio.

$$(\forall t) \quad (atvqualidade(t) \rightarrow atvapoio(t)) \quad (A34)$$

Decomposição de Processo e Atividade

Um **processo** é um conjunto de atividades estruturadas e destinadas a resultar em um artefato ou serviço de valor para a organização ou para um determinado cliente ou mercado. Implica em uma ordenação específica das atividades com começo, fim, insumos e produtos claramente identificados. A figura 5.24 ilustra como processos e atividades são decompostos.

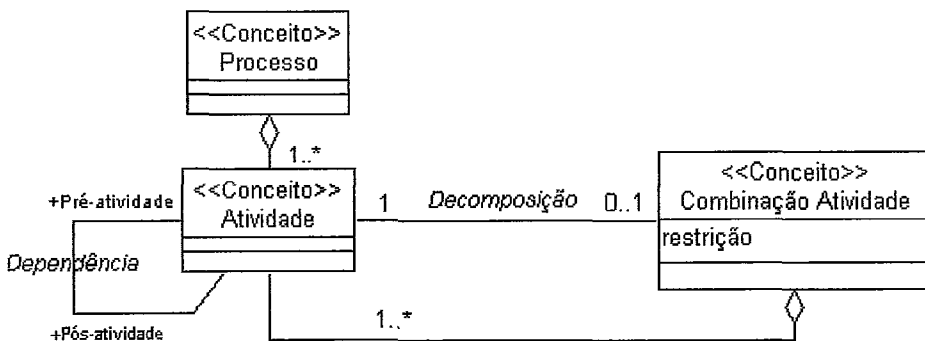


Figura 5.24 – Decomposição de Processo e de Atividade

Uma **combinação de atividade** é um agrupamento de atividades que permite representar a decomposição de uma atividade em um conjunto de atividades e, inclusive, estabelecer uma possível restrição a ser considerada na execução das

atividades agrupadas. Uma restrição pode determinar que as atividades que compõem a combinação devem ser todas obrigatoriamente executadas ou que elas são mutuamente exclusivas.

Uma atividade é dita uma **pré-atividade** de uma outra atividade, se precisa ser executada para que a outra, chamada de **pós-atividade**, também o seja.

Para formalizar as relações ilustradas na figura 5.24, foram definidos os predicados **preatividade**(t_1, t_2) e **posatividade**(t_2, t_1), indicando que t_1 é uma pré-atividade de (antecede) t_2 ou, de forma inversa, que t_2 é uma pós-atividade de (sucede) t_1 . Axiomas similares aos axiomas A8 e A9 foram definidos envolvendo estes predicados. Também foram definidos os predicados **combinação_atividade**(c, r), que indica que c é uma combinação de atividades que obedece à restrição r , onde r assume um dos seguintes valores {*Obrigatoriedade, Exclusividade, Nenhuma*}; **decomposição**(t, c), que indica que a atividade t é uma atividade definida por uma combinação de atividades c ; e, finalmente, **combatv_composição**(t, c), que indica que a atividade t faz parte da combinação de atividades c .

O axioma a seguir estabelece como o predicado **preatividade**(t_1, t_2) é definido em termos dos predicados que representam os insumos e produtos dessas atividades.

$$\boxed{(\forall t_1, t_2) \text{ (preatividade}(t_1, t_2) \leftrightarrow (\exists s) (\text{produto}(s, t_1) \wedge (\text{insumo-1}(s, t_2) \vee \text{insumo-2}(s, t_2))))} \quad \text{(A35)}$$

Também foram definidos os conceitos de **super-atividade** e **sub-atividade**, onde uma super-atividade é uma atividade que não pode ser executada diretamente, isto é, ela é composta de outras atividades menores e sua execução se dá, na realidade, através da execução dessas sub-atividades. A partir destes conceitos, foram definidos os predicados **subatividade**(t_1, t_2), **superatividade**(t_2, t_1), **macroatividade**(t) e **atividadeelementar**(t), o axioma abaixo (A36) e axiomas similares aos axiomas A7 a A11.

$$\boxed{(\forall t_1, t_2) \text{ (subatividade}(t_1, t_2) \leftrightarrow (\exists c) (\text{combinação_atividade}(c, *) \wedge \text{decomposição}(t_2, c) \wedge \text{combatv_composição}(t_1, c)))} \quad \text{(A36)}$$

onde o asterisco (*) indica que não importa o valor atribuído para este argumento¹⁵.

Os insumos e produtos de uma atividade composta são estabelecidos em termos dos insumos e produtos de suas sub-atividades, sendo obtidos através dos seguintes axiomas:

¹⁵ Sempre que, em um axioma, um dos argumentos de um predicado puder assumir um valor arbitrário, este argumento será representado com um asterisco (*).

$$(\forall t, t_1, \dots, t_n, s) (\text{subatividade}(t_1, t) \wedge \dots \wedge \text{subatividade}(t_n, t) \wedge \quad (A37)$$

$$\text{insumo-1}(s, t) \wedge ((\neg \exists t_k) \text{produto}(s, t_k)) \rightarrow \text{insumo-1}(s, t)$$

$$(\forall t, t_1, \dots, t_n, s) (\text{subatividade}(t_1, t) \wedge \dots \wedge \text{subatividade}(t_n, t) \wedge \quad (A38)$$

$$\text{insumo-2}(s, t) \wedge ((\neg \exists t_k) \text{produto}(s, t_k)) \rightarrow \text{insumo-2}(s, t)$$

$$(\forall t, t_1, \dots, t_n, s) (\text{subatividade}(t_1, t) \wedge \dots \wedge \text{subatividade}(t_n, t) \wedge \quad (A39)$$

$$\text{produto}(s, t) \wedge ((\neg \exists t_k) (\text{insumo-2}(s, t_k)) \rightarrow \text{produto}(s, t))$$

onde t_i e $t_k \in \{t_1, t_2, \dots, t_n\}$.

Por fim, o axioma abaixo estabelece a dependência entre os recursos de uma sub-atividade e de sua super-atividade.

$$(\forall t_1, t, b) (\text{recursomaterial}(b, t_1) \wedge \text{subatividade}(t_1, t) \rightarrow \quad (A40)$$

$$\text{recursomaterial}(b, t)$$

$$(\forall t_1, t, c) (\text{recursointelectual}(c, t_1) \wedge \text{subatividade}(t_1, t) \rightarrow \quad (A41)$$

$$\text{recursointelectual}(c, t)$$

Adoção de Procedimento

Procedimentos são condutas bem estabelecidas e ordenadas para a execução de atividades [123]. O aspecto da adoção de procedimento é ilustrado na figura 5.25.

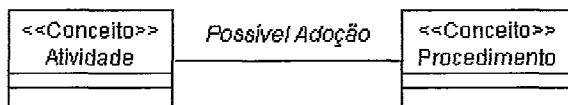


Figura 5.25 – Adoção de Procedimento

Para formalizar o aspecto da adoção de procedimento, foi definido o predicado **possiveladoção(p,t)**, que indica que o procedimento **p** pode ser adotado para a execução da atividade **t**.

Taxonomia de Procedimento

A taxonomia de procedimento (figura 5.26) define como um procedimento pode ser classificado quanto a sua natureza, de acordo com o propósito da ontologia.

Métodos são procedimentos sistemáticos, definindo passos (sub-atividades) e heurísticas para a execução de uma ou mais atividades. Métodos podem ser classificados, de acordo com o tipo de atividades que podem apoiar, em **métodos de gerência**, **métodos de avaliação da qualidade** e **métodos operacionais**.

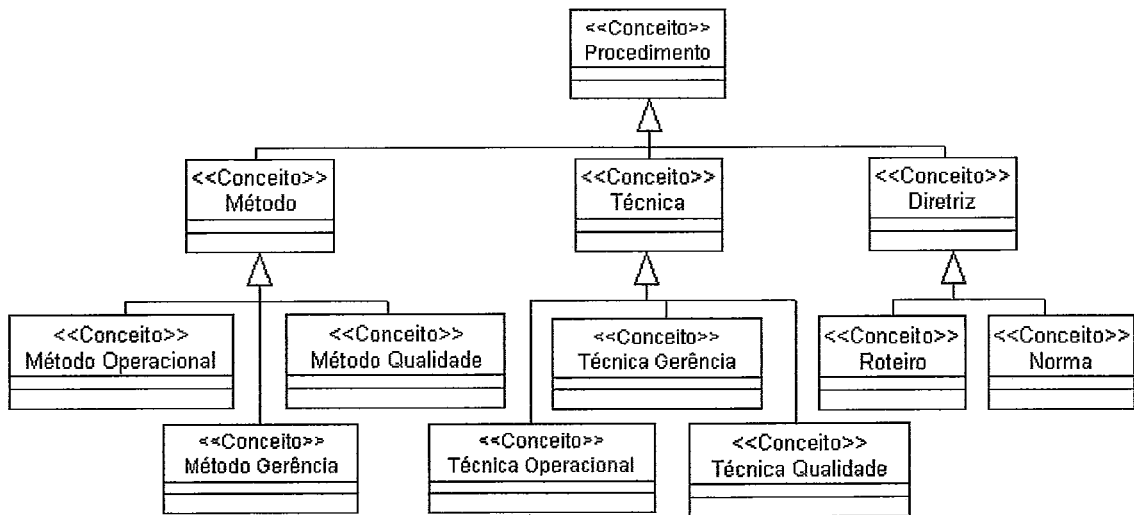


Figura 5.26 – Taxonomia de Procedimento

Técnicas são procedimentos que descrevem apenas aspectos gerais para a execução de uma atividade, não descrevendo como executá-la em termos de sub-atividades. Técnicas podem ser classificadas, de acordo com o tipo de atividades que podem apoiar, em **técnicas de gerência**, **técnicas de avaliação da qualidade** e **técnicas operacionais**.

Diretriz são procedimentos que visam estabelecer um padrão para a execução de atividades. Diretrizes podem ser classificadas em roteiros e normas.

Roteiros são diretrizes para a elaboração de documentos, estabelecendo os principais tópicos a serem abordados. Roteiros, portanto, só podem ser utilizados em atividades que produzam documentos.

Normas são diretrizes que visam estabelecer padrões para a execução ou avaliação de atividades ou processos, o que exclui a elaboração de documentos.

Para formalizar o conceito de procedimento e seus vários tipos, foram definidos os seguintes predicados: **procedimento(p)**, **método(m)**, **técnica(t)**, **diretriz(d)**, **metgerência(m)**, **metqualidade(m)**, **metoperacional(m)**, **tecgerência(t)**, **tecqualidade(t)**, **tecooperacional(t)**, **roteiro(r)** e **norma(n)**.

Como pode ser observado, há uma forte relação entre tipos de procedimentos e tipos de atividades. Roteiros só podem ser adotados por atividades que produzam documentos (axioma A42); métodos e técnicas operacionais, por atividades operacionais (axioma A43); métodos e técnicas de gerência, por atividades de gerência (axioma A44); e métodos e técnicas de avaliação da qualidade, por atividades de avaliação da qualidade (axioma A45).

$(\forall r,t)$	$(roteiro(r) \wedge possíveladoção(r,t) \rightarrow$	(A42)
	$(\exists s) (documento(s) \wedge produto(s,t)))$	
$(\forall p,t)$	$((metoperacional(p) \vee tecoperacional(p)) \wedge$	(A43)
	$possíveladoção(p,t) \rightarrow atvoperacional(t))$	
$(\forall p,t)$	$((metgerência(p) \vee tecgerência(p)) \wedge$	(A44)
	$possíveladoção(p,t) \rightarrow atvgerência(t))$	
$(\forall p,t)$	$((metqualidade(p) \vee tecqualidade(p)) \wedge$	(A45)
	$possíveladoção(p,t) \rightarrow atvqualidade(t))$	

Método como Procedimento Sistemático

Um método, como procedimento sistemático, descreve uma ou mais atividades em termos de outras atividades, o que está representado na figura 5.27.

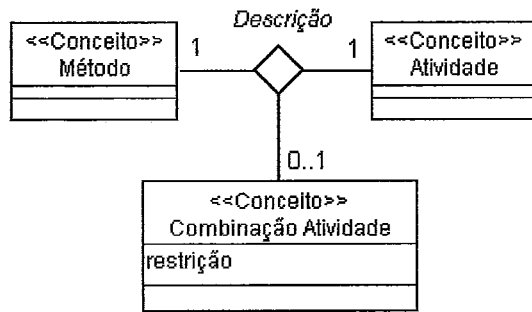


Figura 5.27 – Método como Procedimento Sistemático

Para formalizar o modelo da figura 5.27, foi definido o predicado *descrição(m,t,c)*, indicando que o método *m* descreve a atividade *t* através da combinação de atividades *c*. Além disso, os seguintes axiomas de consolidação devem ser observados:

$(\forall c,m,t)$	$(descrição(m,t,c) \rightarrow$	(A46)
	$possíveladoção(m,t) \wedge combinação_atividade(c,*))$	
$(\forall c,m,t,t_1)$	$(combav_composição(t_1,c) \wedge descrição(m,t,c) \rightarrow$	(A47)
	$t_1 \neq t \wedge \neg superatividade(t_1,t))$	

Sendo assim, um método não pode ser adotado por atividades elementares, assim como técnicas só podem ser adotadas por atividades elementares, uma vez que não descrevem como a atividade deve ser decomposta. Os axiomas, no entanto, devem considerar que o propósito da ontologia é descrever e esta descrição pode estar incompleta, ou seja, um método pode não estar especificado. Os axiomas a seguir refletem estas considerações.

$$\begin{aligned}
 (\forall m,a) \quad & (\text{método}(m) \wedge (\exists c) \text{ descrição}(m,a,c) \rightarrow \text{atividadeelementar}(a)) \quad (A48) \\
 (\forall a,t) \quad & (\text{técnica}(t) \wedge \text{possíveladoção}(t,a) \rightarrow \text{atividadeelementar}(a)) \quad (A49)
 \end{aligned}$$

Automatização de Procedimento

A automatização de procedimento é o aspecto tratado na figura 5.28.

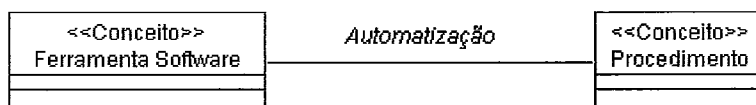


Figura 5.28 – Automatização de Procedimento

Para formalizar o aspecto da automatização de procedimento, foi definido o predicado **automatização(f,p)**, que indica que a ferramenta de software *f* pode ser utilizada para (semi-)automatizar o procedimento *p*. O axioma de consolidação abaixo também foi definido para formalizar a restrição de que uma ferramenta de software só pode ser requerida por uma atividade, se for capaz de (semi-)automatizar um procedimento passível de adoção pela atividade.

$$\begin{aligned}
 (\forall t,f) \quad & ((\text{ferramentasoft}(f) \wedge \text{recursomaterial}(f,t) \rightarrow \text{possíveladoção}(p,t) \wedge \text{automatização}(f,p))) \quad (A50)
 \end{aligned}$$

Processos definidos na Organização e Normas Associadas

Os processos definidos na organização e normas associadas são os aspectos modelados na figura 5.29.

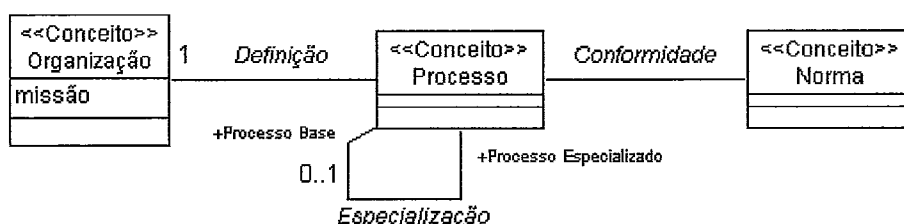


Figura 5.29 – Processos definidos na Organização e Normas Associadas

Para formalizar as relações da figura 5.29, foram definidos os predicados **definição(p,r)**, indicando que *p* é um processo da organização *r*, **conformidade(p,n)**, indicando que o processo *p* está em conformidade com a norma *n*; além de **processoespec(p₁,p₂)** e **processobase(p₂,p₁)**, indicando que *p₁* é um processo especializado de *p₂* ou, de forma inversa, que o processo *p₂* foi utilizado como base para a definição de *p₁*. Também foram definidos axiomas similares aos axiomas A8 e A9 envolvendo estes predicados.

Projetos da Organização

Os projetos da organização são o último aspecto tratado nesta sub-ontologia (figura 5.30).

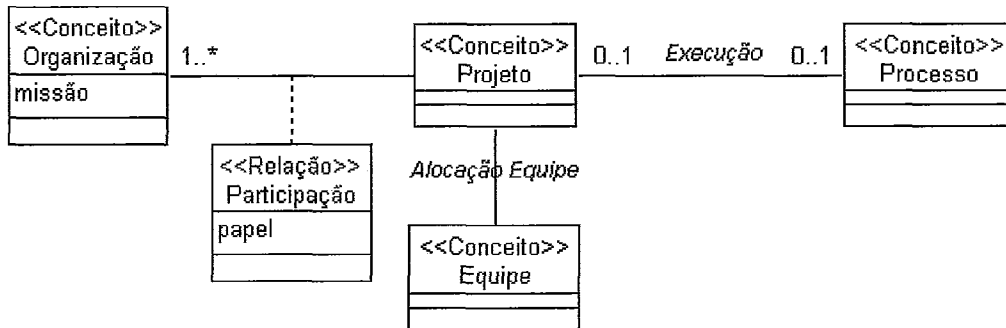


Figura 5.30 – Projetos da Organização

Projetos são empreendimentos que são conduzidos de acordo com determinado esquema, normalmente expresso em um plano do projeto, que define, entre outros, o processo para execução do projeto e a equipe ou as equipes de projeto.

Para formalizar as relações ilustradas na figura 5.30, foram definidos os predicados *participação(r,p,l)*, que indica que a organização *r* participa do projeto *p* exercendo o papel *l*, onde *l* assume um dos seguintes valores {*Cliente, OrgSoftware, Parceiro, Fornecedor*}, *alocação_equipe(q,p)*, que indica que a equipe *q* é responsável pela execução de atividades no (ou seja, está alocada ao) projeto *p*; e *execução(p,c)*, que indica que o projeto *p* é executado segundo o processo *c*.

Uma vez definido o predicado *alocação_equipe(q,p)*, o axioma de consolidação que reflete o conceito de comissão pôde ser, enfim, definido.

$$\boxed{
 \begin{aligned}
 (\forall q) \quad & (equipe(q) \wedge (\neg \exists p) \text{alocação_equipe}(q,p) \quad (A51) \\
 & \leftrightarrow \text{comissão}(q))
 \end{aligned}
 }$$

➤ Sub-ontologia de Estratégia Geral

A sub-ontologia de Estratégia Geral estabelece o vocabulário necessário para descrever os aspectos gerais que definem como a organização interage com o ambiente, respondendo às questões de competência Q1.1 a Q1.4. Com base nessas questões de competência, foi verificada a necessidade de tratar os seguintes aspectos: domínios de atuação, artefatos/serviços oferecidos e relação com organizações clientes.

Domínios de Atuação

A figura 5.31 apresenta a relação que captura o aspecto dos domínios de atuação de uma organização.

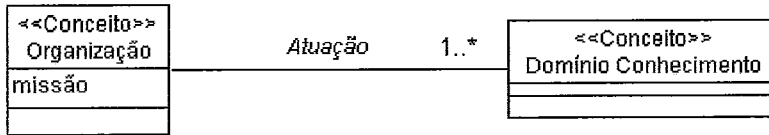


Figura 5.31 – Domínios de Atuação

O predicado **atuaçao(r,d)** foi definido para formalizar a relação acima, indicando que a organização *r* atua no domínio de conhecimento *d*, ou seja, que possui capital intelectual relativo ao domínio de conhecimento e executa atividades que requerem conhecimentos que pertencem a este domínio.

Artefatos e Serviços oferecidos pela Organização

Na figura 5.32, o aspecto dos artefatos/serviços oferecidos pela organização é abordado.

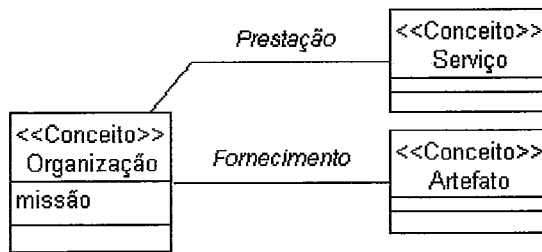


Figura 5.32 – Artefatos/Serviços Oferecidos

Um **serviço** é oferecido para satisfazer a necessidade ou desejo de um determinado cliente ou mercado, sendo não tangível e abstrato, em contraste com o artefato, que é um elemento tangível.

Para formalizar as relações ilustradas na figura 5.32, foram definidos o predicado **prestação(r,s)**, que indica que a organização *r* presta o serviço *s* para terceiros, e o predicado **fornecimento(r,s)**, que indica que a organização *r* fornece o artefato *s* para terceiros.

Relação com Organizações Clientes

Por fim, a figura 5.33 mostra como foi tratado o aspecto da relação da organização com suas organizações clientes.

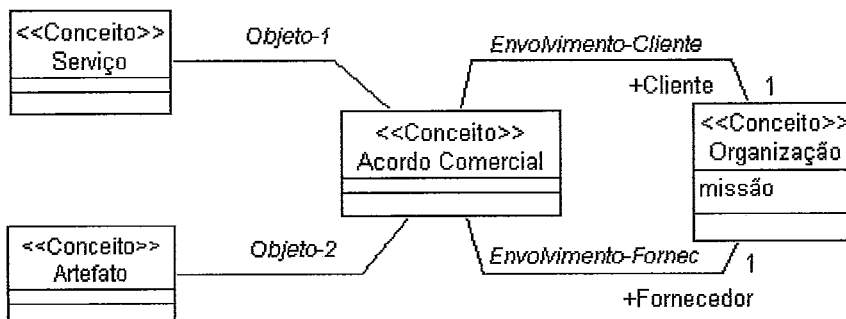


Figura 5.33 – Relação com Organizações Clientes

Um **acordo comercial** é um instrumento que estabelece uma relação de negócio entre duas organizações, que atuam, respectivamente, como fornecedora e cliente.

Para formalizar o modelo da figura 5.33, foram definidos os predicados: **envolvimento_cliente(c,r)**, que indica que a organização *c* participa do acordo comercial *r* como organização cliente; **envolvimento_fornec(f,r)**, que indica que a organização *f* participa do acordo comercial *r* como fornecedora de artefatos e/ou serviços; **objeto-1(s,r)**, que indica que o serviço *s* é objeto do acordo comercial *r*, sendo um serviço prestado à organização cliente; e **objeto-2(s,r)**, que indica que o artefato *s* é objeto do acordo comercial *r*, sendo fornecido à organização cliente. Os axiomas abaixo completam a formalização.

$$(\forall c,f) \quad (\text{cliente}(c,f) \leftrightarrow (\exists r) (\text{envolvimento_cliente}(c,r) \wedge \text{envolvimento_fornec}(f,r))) \quad (\text{A52})$$

$$(\forall f,r,s) \quad ((\text{envolvimento_fornec}(f,r) \wedge \text{objeto-1}(s,r)) \rightarrow \text{prestação}(f,s)) \quad (\text{A53})$$

$$(\forall f,r,s) \quad ((\text{envolvimento_fornec}(f,r) \wedge \text{objeto-2}(s,r)) \rightarrow \text{fornecimento}(f,s)) \quad (\text{A54})$$

$$(\forall c,r,s) \quad ((\text{envolvimento_cliente}(c,r) \wedge \text{objeto-1}(s,r)) \rightarrow \text{demanda_serviço}(c,s)) \quad (\text{A55})$$

$$(\forall c,r,s) \quad ((\text{envolvimento_cliente}(c,r) \wedge \text{objeto-2}(s,r)) \rightarrow \text{demanda_artefato}(c,s)) \quad (\text{A56})$$

5.8 Conclusão

Este capítulo apresentou um modelo que especifica os componentes de um ADSOrg e também sugeriu formalismos para representação dos conhecimentos considerados mais básicos, servindo de ponto de partida para a construção de tais ambientes.

Capítulo 6 – Pesquisa sobre a Importância dos Tipos de Conhecimento

6.1 Introdução

No capítulo 1 foi mencionado que a Engenharia de Software é um domínio de conhecimento amplo e que vários tipos de conhecimento são requeridos durante o desenvolvimento e a manutenção de software. O capítulo 2 destacou a importância de fornecer, desde o início, conhecimento útil na Memória Organizacional. Desta forma, verificou-se a necessidade de identificar quais conhecimentos os desenvolvedores consideram mais importantes, para que estes possam ser priorizados nos esforços de incorporação de conteúdo e desenvolvimento de serviços e ferramentas para ADSOrg.

Para atender a essa necessidade, uma pesquisa foi realizada em 2002 com programadores, analistas, gerentes de projeto e diretores de empresas sediadas em Salvador que desenvolvem e/ou dão manutenção em software. O processo utilizado para realização da pesquisa foi adaptado do processo de experimentação proposto por WOHLIN *et al.* [194] de acordo com princípios fornecidos por KITCHENHAM e PFLEEGER [195,196,197,198,199,200]. Neste capítulo, as informações relativas a cada fase da pesquisa são fornecidas conforme as recomendações de WOHLIN *et al.* [194] para apresentação e empacotamento dos resultados de um experimento.

6.2 Definição da Pesquisa

O objetivo definido para a pesquisa, seguindo o modelo GQM [106] para definição de metas, foi:

Analisar os ambientes organizacionais para desenvolvimento e manutenção de software

Com o propósito de caracterizar os conhecimentos mencionados na literatura técnica de Engenharia de Software

Com respeito à importância percebida em termos da necessidade de tê-los, de alguma forma, disponíveis ao longo do desenvolvimento e da manutenção de software

Do ponto de vista dos programadores, analistas, gerentes de projeto e diretores

No contexto dos projetos de software conduzidos no ano de 2002 em empresas sediadas em Salvador

A pesquisa com objetivo mais próximo do descrito acima foi realizada por LETHBRIDGE em 1998 [201]. Os objetivos da pesquisa foram: 1) entender quais eram os tópicos educacionais mais importantes para os profissionais da área de software ao longo de suas carreiras e 2) identificar os tópicos nos quais os conhecimentos ou educação desses profissionais podiam ser melhorados. O relato feito por LETHBRIDGE [201] forneceu parâmetros que ajudaram no planejamento da pesquisa aqui descrita, tais como: número de perguntas e formas de aplicação do questionário, tempo médio de resposta e duração da etapa de operação da pesquisa.

6.3 Planejamento da Pesquisa

O planejamento da pesquisa envolveu: a identificação da população alvo, a seleção da amostra, a formulação das hipóteses, a seleção do projeto mais adequado, a instrumentação da pesquisa e a avaliação de sua validade. Os resultados de cada uma destas atividades são descritos a seguir.

6.3.1 Identificação da População Alvo

A população alvo estabelecida para a pesquisa foi a força de trabalho envolvida com o desenvolvimento e a manutenção de software na cidade de Salvador, que foi estimada em 472 pessoas em 2002. Este número foi obtido a partir dos dados da Relação Anual de Informações Sociais (RAIS) - Ano Base 2001, fornecidos pelo Ministério do Emprego e Trabalho.

Segundo os dados da RAIS, existiam 32 empresas que tinham como ocupação o desenvolvimento e a manutenção de software na cidade de Salvador em 31 de dezembro de 2001. O número de empresas no estado da Bahia e no Brasil eram, respectivamente, 53 e 2.640. Ainda segundo a RAIS, o número de empregos formais nas empresas de Salvador era de 938, dos quais 472 referentes a ocupações indefinidas ou que puderam ser consideradas como envolvidas no desenvolvimento e manutenção de software. A respeito deste número, é importante ressaltar que ele não inclui bolsistas e estagiários e que poucos diretores foram informados. Isto significa que o tamanho exato da população não era 472, até mesmo porque existiam e existem empresas que desenvolvem e mantêm software sem ter estas atividades como ocupação, mas este foi o número mais próximo que pôde ser obtido.

6.3.2 Seleção da Amostra

Como não foi requerida nenhuma característica especial para participar da pesquisa, foi estabelecido que pessoas com diferentes níveis de conhecimento e

experiência seriam agrupadas em programadores, analistas, gerentes e diretores. O tamanho inicial da amostra foi de 99 programadores, 203 analistas, 61 gerentes e 30 diretores, totalizando 393 pessoas que representavam a força total de trabalho envolvida no desenvolvimento e na manutenção de software nas 18 empresas que aceitaram participar da pesquisa, incluindo estagiários e bolsistas.

A amostra da população foi selecionada a partir das 27 empresas que responderam às pesquisas sobre Qualidade e Produtividade no Setor de Software Brasileiro nos anos de 1999 e 2001 [202,203] ou que participaram do censo, realizado em 2001, das empresas associadas a SOFTEX [204]. Destas empresas, 6 já haviam sido fechadas ou não puderam ser localizadas. Segundo o manual da RAIS, as 21 empresas localizadas teriam que ter entregue a declaração anual em 2001, mas uma delas não tinha, como ocupação, o desenvolvimento e a manutenção de software e, conseqüentemente, não teve os seus dados incluídos na estimativa da população. Dentre as 21 empresas localizadas, 3 não aceitaram participar da pesquisa alegando período conturbado de trabalho.

6.3.3 Formulação das Hipóteses

Inicialmente, decidiu-se investigar se a importância percebida dos tipos de conhecimento nos ambientes organizacionais para desenvolvimento e manutenção de software depende de fatores como a formação e experiência dos profissionais, tamanho das empresas e tamanho dos projetos. Desta forma, foram formuladas as seguintes hipóteses, todas considerando a opinião de profissionais que desempenham a mesma função em tais ambientes:

- Hipótese Nula (H0): Não é possível identificar conhecimentos cuja importância percebida nos ambientes organizacionais para desenvolvimento e manutenção de software independe da formação e experiência dos profissionais, do tamanho das empresas ou do tamanho dos projetos;
- Hipótese 1 (H1): É possível identificar tipos de conhecimento cuja importância percebida nos ambientes organizacionais para desenvolvimento e manutenção de software independe da formação e experiência de seus profissionais;
- Hipótese 2 (H2): É possível identificar tipos de conhecimento cuja importância percebida nos ambientes organizacionais para desenvolvimento e manutenção de software independe do tamanho das empresas nas quais estão inseridos;
- Hipótese 3 (H3): É possível identificar tipos de conhecimento cuja importância percebida nos ambientes organizacionais para desenvolvimento e manutenção de software independe do tamanho dos projetos sendo conduzidos.

No entanto, acreditando que os fatores mencionados realmente influenciam a percepção da importância dos tipos de conhecimento nos ambientes para desenvolvimento de software, também foi planejada a identificação dos tipos de conhecimentos considerados mais importantes e menos importantes de acordo com cada um desses fatores.

➤ **Projeto da Pesquisa**

Considerando o objetivo, a população alvo, a amostra e as hipóteses da pesquisa, concluiu-se que esta:

- seria descritiva exploratória, por buscar investigar a importância de uma série de conhecimentos e a sua independência com relação a determinados fatores;
- seria do tipoquete, possuindo amostragem casual não probabilística, o que significava que a probabilidade de participação de um indivíduo da população na pesquisa era desconhecida, dependendo da empresa, através de seu representante, aceitar participar da pesquisa e dos indivíduos efetivamente atenderem à solicitação de participação;
- possuiria projeto do tipo corte transversal com controle de caso, o que significava que os participantes seriam questionados, em um ponto particular do tempo, sobre suas experiências passadas.

➤ **Instrumentação**

Para instrumentação da pesquisa, um questionário foi elaborado e refinado com a colaboração de três pesquisadores sênior da área de Engenharia de Software e quatro estudantes de mestrado da COPPE/UFRJ.

Com o objetivo de avaliar a instrumentação e a efetividade dos procedimentos de acompanhamento, o questionário foi aplicado, de forma não supervisionada, a um grupo de seis pessoas (dois gerentes, dois analistas e dois programadores) de uma das empresas que iriam participar da pesquisa. Estas pessoas foram solicitadas por um membro de sua equipe de projeto, não necessariamente superior hierarquicamente, a preencher o questionário e identificar potenciais problemas. Nenhuma melhoria foi sugerida e nenhum problema foi observado no questionário. O tempo médio de resposta registrado neste momento foi de 26 minutos.

Além de 19 questões para caracterização do profissional, do seu projeto e de sua empresa, o questionário continha 96 questões para coletar a opinião do profissional sobre a importância de vários tipos de conhecimento mencionados na literatura técnica de Engenharia de Software. As questões utilizadas para a coleta de opinião foram compiladas a partir dos tópicos abordados em [140,205,206,207],

enquanto algumas das questões utilizadas para caracterização foram elaboradas a partir de [203,208]. Apesar de 96 ser um grande número de questões, todas podiam ser respondidas em termos da mesma escala, o que facilitou e agilizou a resposta ao questionário.

Logo no início da aplicação do questionário na primeira empresa, observou-se, entretanto, que desenvolvedores de um mesmo projeto estavam considerando diferentes empresas como clientes, fornecedores e parceiros técnicos, apesar dos respectivos conceitos estarem descritos no questionário. Ao mesmo tempo, foi verificado que as perguntas referentes à caracterização do projeto e à caracterização da empresa não precisavam ser respondidas por todos e, mais do que isso, deviam ser respondidas uma única vez, de forma a evitar inconsistências. Para tornar isto possível, o questionário inicial foi dividido em três partes (anexo I), sem a necessidade de reformulação de questões, mas com a inclusão do nome do projeto e da empresa, para permitir a associação de informações, e de três questões opcionais, que, apesar de não fazerem parte da pesquisa, foram adicionadas com a finalidade de possibilitar uma visão uniforme sobre quem eram os clientes, fornecedores e parceiros técnicos. Com a divisão, o questionário principal passou a conter apenas as questões para caracterização do profissional que está respondendo a pesquisa e as 96 questões para a coleta da sua opinião, enquanto os outros questionários passaram a solicitar a caracterização da empresa e a caracterização do projeto. O procedimento estabelecido foi solicitar, ao representante da empresa, o preenchimento do questionário para caracterização da empresa; dos gerentes de projeto, o preenchimento dos questionários de caracterização dos respectivos projetos; e de todos os envolvidos no desenvolvimento e manutenção de software, o preenchimento do questionário principal.

A alteração da instrumentação só foi possível por ter sido efetuada antes dos primeiros profissionais terminarem de responder ao questionário, ainda na primeira empresa em que a pesquisa estava sendo aplicada, e por estar relacionada a questões de caracterização que iriam influenciar apenas a resposta dada às últimas questões para coleta de opinião (questões sobre clientes, fornecedores e parceiros técnicos), estabelecendo um intervalo de tempo suficiente para que a necessidade de alteração fosse percebida e a alteração fosse efetuada. Cabe ressaltar que a alteração da instrumentação não afetou a forma de análise dos dados e nem mesmo a forma de agregação dos mesmos.

O tempo médio de resposta estimado para o questionário principal foi de 23 minutos, obtido a partir do tempo de resposta de 4 participantes.

➤ Avaliação da Validade

As ameaças à validade da pesquisa foram identificadas [194] e procedimentos para tratamento dessas ameaças foram adotados conforme descrito na tabela 6.1.

Tabela 6.1 – Tratamento das Ameaças à Validade da Pesquisa

Ameaças à Validade da Construção	
Ameaça	Tratamento Adotado
Explicação inadequada das construções	Revisão do planejamento, verificando se todos os termos utilizados eram conhecidos ou estavam definidos
Confusão entre construção e níveis de construção	Revisão do planejamento, verificando as construções nas quais era importante indicar os níveis de presença e não apenas a presença ou ausência
Preocupação das pessoas em serem avaliadas	Explicação dos objetivos da pesquisa e da confidencialidade das respostas aos participantes da pesquisa
Influência da expectativa do pesquisador nos resultados	Revisão do questionário, observando a formulação das perguntas e as possibilidades de resposta fornecidas aos participantes
Ameaças à Validade Interna	
Ameaça	Tratamento Adotado
Má instrumentação	Revisão do questionário quanto ao formato e à formulação das perguntas; avaliação do questionário por outros pesquisadores e teste do mesmo com pessoas que possuíam a mesma formação e experiência dos participantes da pesquisa
Efeito da natural variação do desempenho humano	Não seleção dos participantes de acordo com a sua disponibilidade, desejo de participar da pesquisa, experiência ou formação. Ao contrário, a importância da participação de todos foi enfatizada. No entanto, as empresas foram selecionadas pelo fato de terem aceitado participar anteriormente de outras pesquisas/censos
Ameaças à Validade Externa	
Ameaça	Tratamento Adotado
Seleção não representativa da população	Solicitação de que todos os envolvidos em projetos de software nas empresas selecionadas participassem da pesquisa, buscando-se diversidade e taxa de resposta satisfatória. No entanto, a taxa de resposta foi baixa em algumas categorias e novos estudos precisam ser realizados.
Ameaças à Validade da Conclusão	
Ameaça	Tratamento Adotado
Baixo poder estatístico	Consultoria de um especialista para a escolha dos testes estatísticos adequados para análise dos dados
Violação das condições necessárias para uso dos testes estatísticos	Consultoria de um especialista e uso de programa estatístico que verifica se as condições necessárias para uso dos testes foram satisfeitas
Busca de um resultado específico	Inclusão de tipos de conhecimento no questionário independente dos interesses de trabalho das pessoas envolvidas; fornecimento de espaço para os participantes incluírem tipos de conhecimento não previstos; possibilidade dos participantes não avaliarem a importância de um tipo de conhecimento previsto
Confiabilidade das medições	Revisão do questionário quanto ao formato e conteúdo, buscando assegurar o correto entendimento dos termos e o fornecimento das definições necessárias, além de solicitação de que os participantes considerassem cuidadosamente as respostas fornecidas

6.4 Operação da Pesquisa

Buscando-se alta taxa de resposta e início imediato, a pesquisa começou a ser aplicada nas empresas de forma supervisionada, utilizando-se questionários impressos. No entanto, esta estratégia não pôde ser aplicada em todas as empresas por dois motivos: dificuldade de conciliação de horários para a aplicação da pesquisa e dificuldade de deslocamento dos profissionais alocados em empresas clientes. A solução foi o desenvolvimento de um sistema para aplicação de questionários através da Web, mas, mesmo assim, algumas empresas solicitaram que a pesquisa ainda pudesse ser aplicada de forma não supervisionada com o questionário sendo enviado por fax ou por e-mail. O anexo II apresenta o modelo de classes e algumas telas do sistema para aplicação de questionários na Web, o qual permite agrupamento de perguntas em vários níveis, a especificação de tipo e valores de resposta válidos, a especificação de resposta única ou não, além de complemento textual para respostas pré-definidas.

Desta forma, os preparativos para a execução da pesquisa em uma empresa passaram a ser: obter a indicação da melhor forma de aplicar os questionários na empresa e, a depender da forma escolhida, tomar as medidas necessárias para possibilitar a aplicação do questionário. Isso implicava em entrar em contato com responsável pela área de software para explicar o objetivo e a importância da pesquisa, a instituição responsável, como a empresa tinha sido selecionada para participar, o tempo médio necessário para responder aos questionários e como a confidencialidade das informações obtidas seria preservada. As formas possíveis para aplicação do questionário já foram discutidas, sendo que o sistema foi utilizado por 8 empresas, a aplicação não supervisionada foi adotada em 5 empresas e outras 5 empresas optaram pela aplicação supervisionada.

A taxa de não resposta foi de 55,56% para programadores, 62,56% para analistas, 44,26% para gerentes de projeto e 66,67% para diretores, o que significa que 44 programadores, 76 analistas, 34 gerentes e 10 diretores efetivamente participaram da pesquisa.

Os questionários foram examinados quanto à consistência e completude assim que recebidos, tendo sido percebido que as inconsistências se manifestaram principalmente na caracterização do projeto e na classificação da interação do participante com os clientes, fornecedores e parceiros técnicos envolvidos no projeto. Os participantes, cujos questionários apresentavam inconsistências e/ou questões em branco, foram solicitados a esclarecerem as inconsistências e/ou responderem às questões em branco. Questionários que permaneceram com questões em branco não

foram eliminados. Foram eliminados 11 questionários: 5 por apresentarem uma única resposta para todos os itens; 2 pelos participantes terem respondido do ponto de vista da empresa cliente, onde estavam alocados; 1 por vários itens constarem como “Não Avaliado” e ainda haver inconsistências; 3 pelos participantes demonstrarem cansaço, percebido por respostas dadas em bloco a partir de um determinado ponto do questionário.

A amostra válida foi, então, de 41 programadores, 70 analistas, 32 gerentes e 10 diretores, totalizando 153 questionários. Todos autorizaram o uso das informações coletadas na pesquisa desde que a sua identificação e a identificação da sua empresa não fossem divulgadas (anexo I).

6.5 Análise dos Dados da Pesquisa

Grupos de empresas, projetos e participantes foram estabelecidos para permitir o teste das hipóteses. As empresas foram classificadas em Micro, Pequena, Média e Grande, conforme mostra a tabela 6.2, obtida de [203]. Os projetos foram divididos em Pequeno, Médio, Grande e de Manutenção (tabela 6.3). A tabela 6.3 foi obtida a partir da análise dos dados coletados na própria pesquisa, uma vez que não foi encontrada uma classificação de projetos quanto ao esforço que fosse adequada e amplamente utilizada.

Tabela 6.2 – Grupo de Empresas

Sigla	Grupo	Critério (n° de pessoas)
μE	Micro	Menos de 10
PE	Pequena	de 10 a 49
ME	Média	de 50 a 99
GE	Grande	100 ou mais

Tabela 6.3 – Grupo de Projetos

Sigla	Grupo	Critério (homens-hora)
PP	Pequeno	menos de 2500
MP	Médio	de 2500 a 16999
GP	Grande	17000 ou mais
mP	Manutenção	duração indeterminada

Para a construção da tabela 6.3, foi analisada a distribuição do esforço atribuído aos projetos de software. O esforço foi calculado multiplicando-se: número médio de pessoas na equipe, duração conhecida do projeto em meses, número de dias úteis em um mês e número de horas de trabalho por dia. Foram adotados 22 como número de dias úteis em um mês e 9 como número de horas de trabalho por dia, pois várias empresas da amostra estão registradas como estabelecimentos comerciais e adotam a jornada de trabalho de 44 horas semanais. No entanto, alguns projetos possuíam pessoas alocadas por tempo indeterminado para executar as atividades necessárias à manutenção de um sistema, não sendo possível calcular o esforço envolvido nestes projetos. Desta forma, a categoria *Manutenção* foi criada para agrupar tais projetos, que não foram considerados na distribuição de esforço. Considerando a distribuição de esforço para os demais projetos, verificou-se que o 25%-percentil da distribuição, o

esforço de 2.475 homens-hora, era muito próximo do valor de esforço limítrofe para projetos pequenos utilizado por uma das classificações encontradas na Web [209] e que o 75%-percentil da distribuição, o esforço de 16.929 homens-hora, era bastante inferior ao valor de esforço limítrofe para projetos médios utilizado pela mesma classificação. Buscando-se equilibrar o número de projetos em cada grupo e reconhecendo-se que projetos considerados grandes nos Estados Unidos podem ser efetivamente maiores do que os projetos considerados grandes no Brasil, adotou-se os valores 2.500 e 17.000 como valores limítrofes na tabela 6.3, que são, respectivamente, o arredondamento do 25%-percentil e do 75%-percentil da distribuição do esforço atribuído aos projetos de software.

Já os participantes da pesquisa foram agrupados em programadores, analistas, gerentes e diretores, onde o grupo de gerentes inclui gerentes de projeto, de contrato e de área, e o grupo de diretores inclui diretores, assessores da diretoria e empresários. Em cada um dos grupos, pesos foram atribuídos à opinião dos participantes de acordo com a formação e o tempo de experiência dos mesmos, utilizando-se a seguinte fórmula:

$$\text{Peso (i)} = \frac{\text{Tempo de Experiência (i)} + \text{Função de Escolaridade (i)}}{\text{Mediana (Tempo de Experiência + Função de Escolaridade)}}$$

A Função de Escolaridade tem seu valor calculado pela distância relativa entre a escolaridade do participante e a menor escolaridade possível (2º grau) em termos da quantidade de anos necessários para obtê-las e da utilidade da escolaridade do participante para o exercício de sua função na empresa. Foi assumido que graduação não concluída em Informática representa 2 anos úteis para o exercício de qualquer uma das funções consideradas; e que graduação, ou mesmo especialização, em outras áreas representam 2 anos úteis para o exercício das funções de analista, gerente ou diretor, mas, para programador, só representam 2 anos quando forem na área de Ciências Exatas. A tabela 6.4 explicita todos os valores assumidos pela Função de Escolaridade. Os valores da mediana para os grupos de programadores, analistas, gerentes e diretores foram, respectivamente, 5, 9, 13,5 e 17.

Tabela 6.4 – Função de Escolaridade

	2º Grau	Em Grad. Informática	Grad./Espec. Outra Área	Grad. Informática	Espec. Informática	Mestrado Informática
Programador	0	2	2* 0	4	5	6
Analista/Gerente/Diretor	0	2	2	4	5	6

* Graduação na Área de Ciências Exatas

Cada um dos grupos de participantes ainda foi dividido em 3 subgrupos de acordo com a proximidade dos pesos obtidos, o que pode ser observado nos gráficos apresentados nas figuras 6.1 a 6.4. O objetivo foi obter grupos mais homogêneos em termos de formação e tempo de experiência. A tabela 6.5 caracteriza os subgrupos.

Tabela 6.5 – Subgrupos de Participantes

Programador	Faixa de Pesos	Analista	Faixa de Pesos	Gerente	Faixa de Pesos	Diretor	Faixa de Pesos
P1	0,44 a 0,80	A1	0,33 a 0,94	G1	0,37 a 0,74	D1	0,59 a 0,71
P2	1,00 a 1,40	A2	1,00 a 1,44	G2	0,89 a 1,26	D2	1,00 a 1,18
P3	1,80 a 2,80	A3	1,56 a 2,33	G3	1,33 a 1,70	D3	1,35 a 1,88

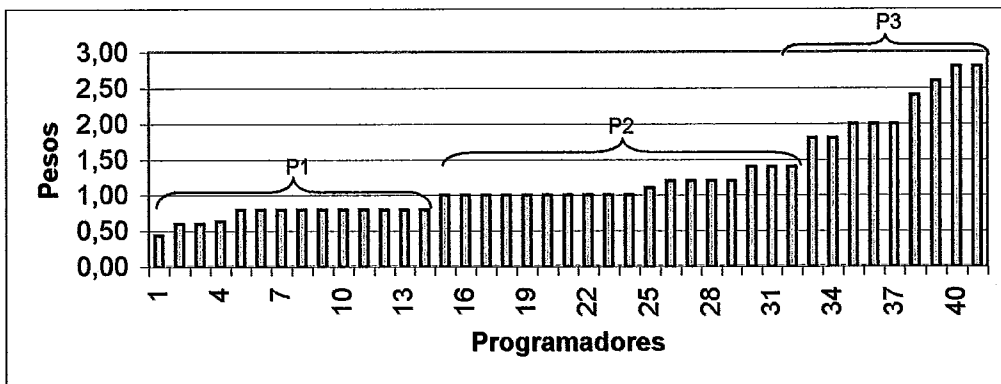


Figura 6.1 – Pesos dos Participantes Programadores

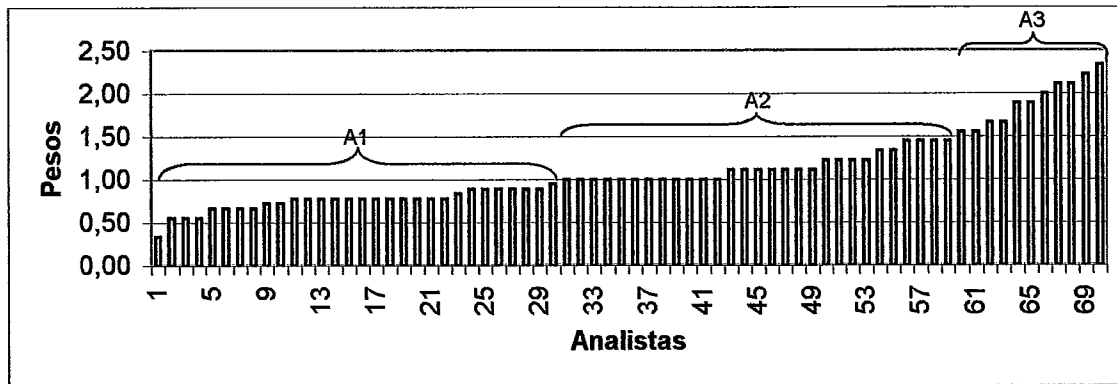


Figura 6.2 – Pesos dos Participantes Analistas

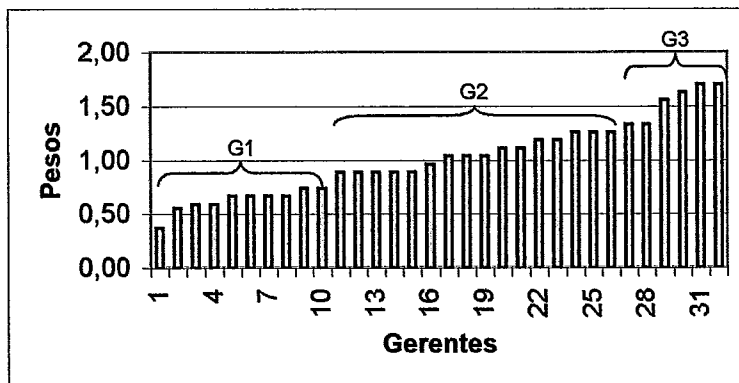


Figura 6.3 – Pesos dos Participantes Gerentes

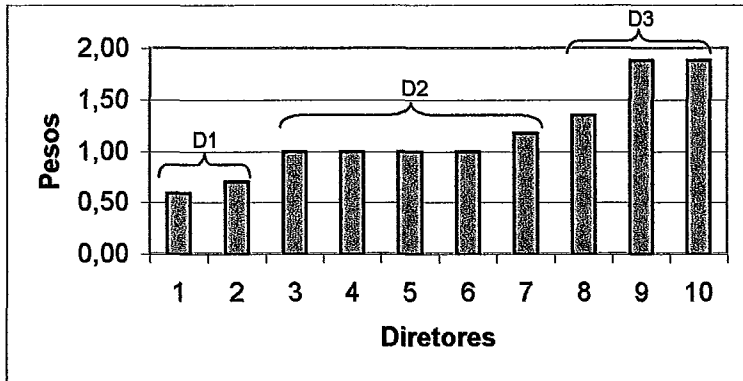


Figura 6.4 – Pesos dos Participantes Diretores

A análise descritiva dos dados demográficos é apresentada nas figuras 6.5 a 6.16.

As figuras 6.5 a 6.7 ilustram a distribuição das empresas participantes da pesquisa quanto ao tamanho da empresa, atividades de desenvolvimento de software executadas na empresa e rotatividade na área de software. Na figura 6.5, o tipo de empresa *Software* indica que as atividades de desenvolvimento e manutenção de software são atividades de negócio da empresa e o tipo de empresa *Não Software* indica que estas são atividades de apoio ao negócio. Além disso, as empresas foram classificadas quanto ao tamanho com base na tabela 6.2. Na figura 6.6, a categoria *Uso Próprio* significa que a empresa desenvolve e mantém software para uso próprio, o mesmo aplicando-se às demais categorias. Tanto as categorias da figura 6.6 quanto as da figura 6.7 foram obtidas diretamente do questionário para caracterização da empresa.

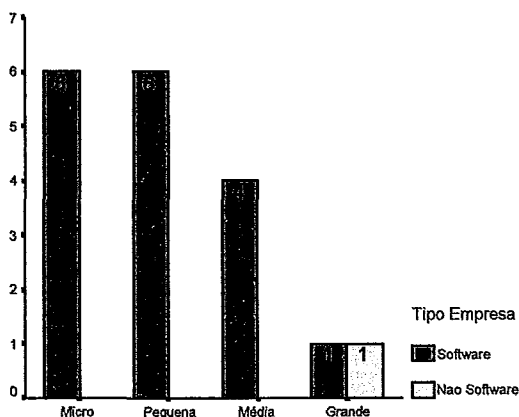


Figura 6.5 – Tamanho das Empresas

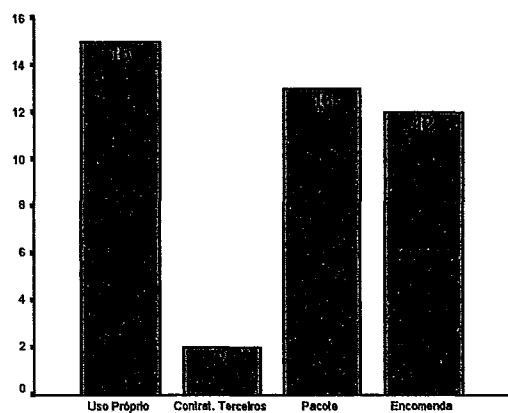


Figura 6.6 – Atividades de Desenvolvimento de Software nas Empresas

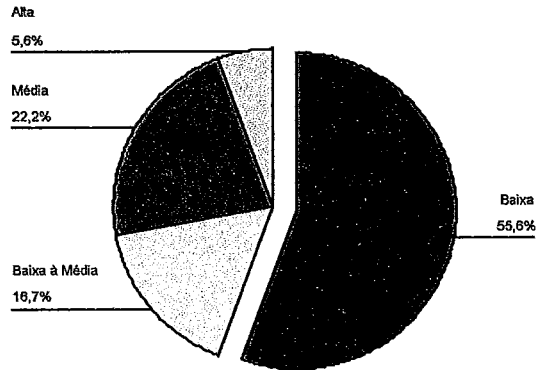


Figura 6.7 – Rotatividade na Área de Software

As figuras 6.8 a 6.10 ilustram a distribuição dos projetos quanto ao tamanho, classe e rotatividade no projeto. Os projetos foram classificados quanto ao tamanho com base na tabela 6.3, sendo 89 o número total de projetos (figura 6.8). Tanto as categorias da figura 6.9 quanto as da figura 6.10 foram obtidas diretamente do questionário para caracterização do projeto, sendo que apenas as principais classes de projeto foram consideradas na análise descritiva por serem suficientes para classificação, de forma única, de todos os projetos.

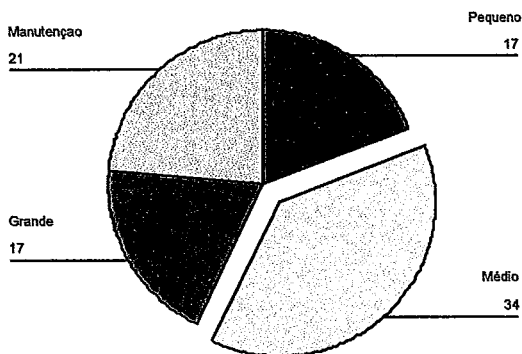


Figura 6.8 – Tamanho dos Projetos

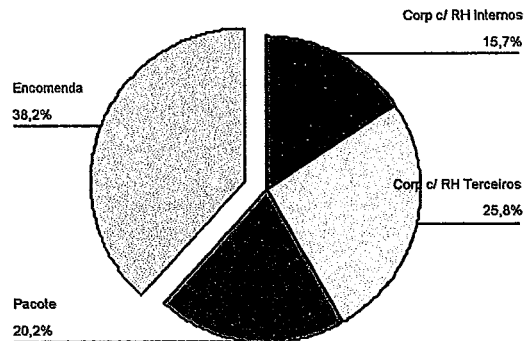


Figura 6.9 – Classe dos Projetos

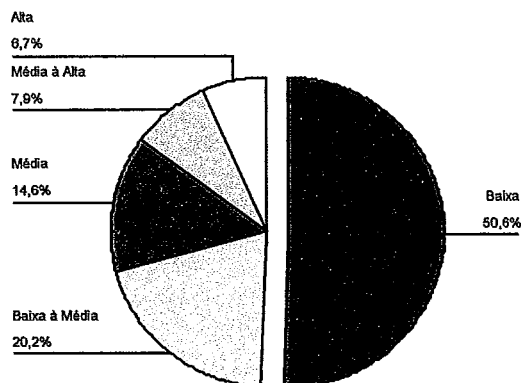


Figura 6.10 – Rotatividade no Projeto

As figuras 6.11 a 6.16 referem-se à caracterização dos profissionais que responderam ao questionário principal da pesquisa. A figura 6.11 evidencia a distribuição dos profissionais de acordo com a função desempenhada e a experiência possuída, totalizando 153 profissionais. A figura 6.12 evidencia a distribuição dos profissionais de acordo com a função desempenhada e a escolaridade. A figura 6.13 ilustra a distribuição dos profissionais de acordo com os grupos estabelecidos na tabela 6.5, ou seja, em termos da função, escolaridade e tempo de experiência dos profissionais. As figuras 6.14 a 6.16 mostram como os profissionais se relacionam com as outras empresas envolvidas no projeto, evidenciando, em termos gerais, a pouca interação com fornecedores e a freqüente interação com os clientes.

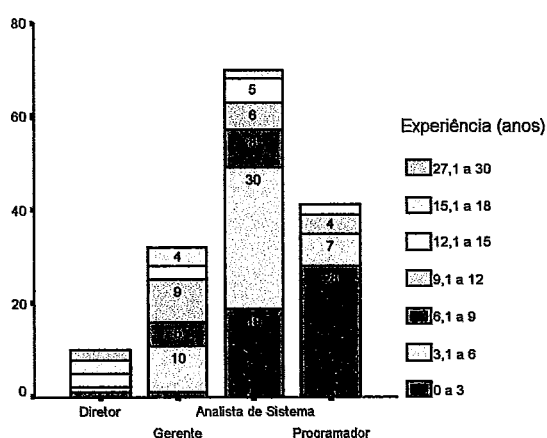


Figura 6.11 – Tempo de Experiência

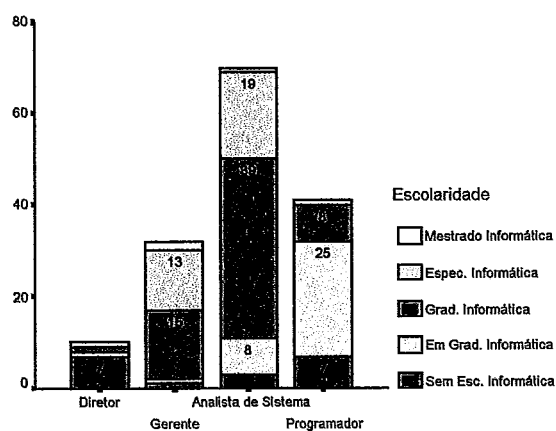


Figura 6.12 – Escolaridade

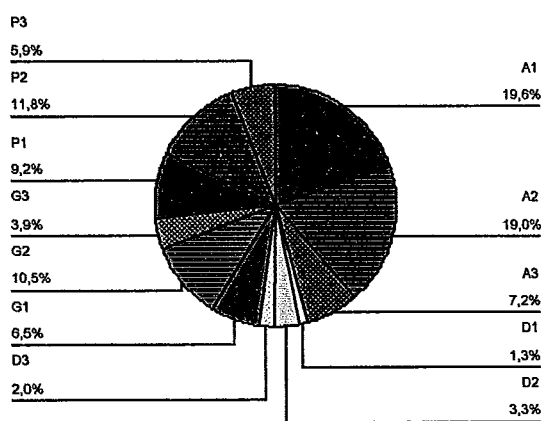


Figura 6.13 – Grupo de Participantes

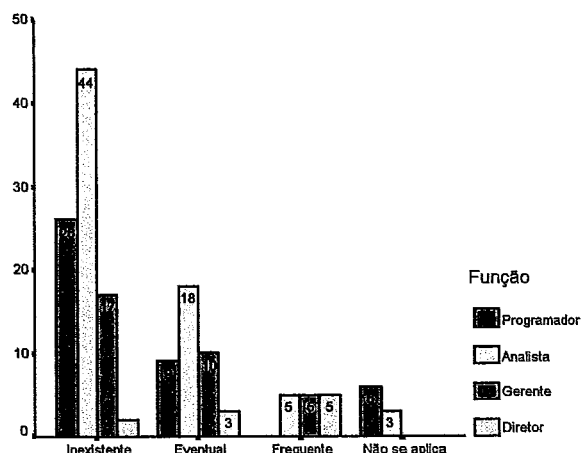


Figura 6.14 – Interação com Fornecedores

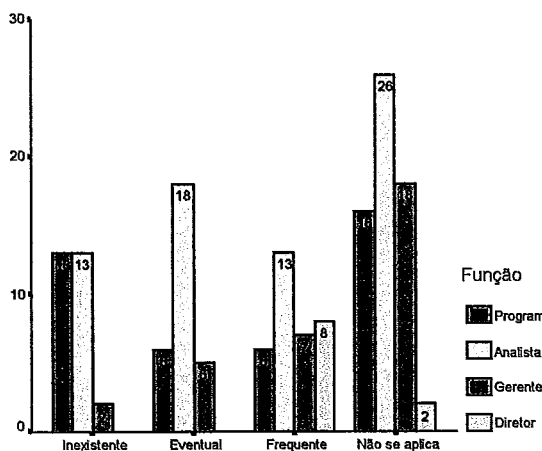


Figura 6.15 – Interação com Parceiros Técnicos

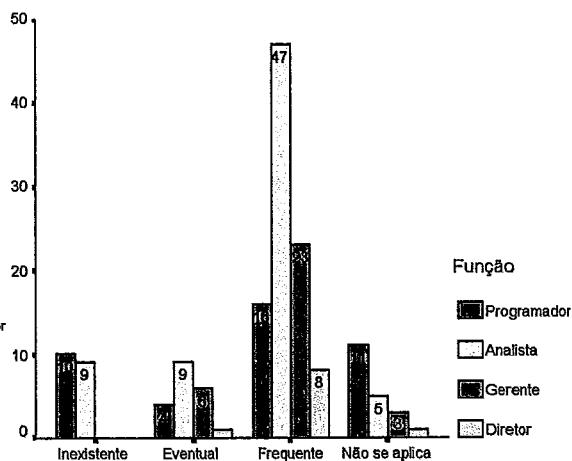


Figura 6.16 – Interação com Clientes

Para testar as hipóteses, foi selecionado o teste não paramétrico¹ do Qui-quadrado de Pearson, que baseia-se em dados na forma de freqüência e pode ser utilizado para investigar se os dados provenientes de dois ou mais grupos apresentam a mesma distribuição de freqüência² [194], o que significa que os dados obtidos são independentes dos grupos específicos. No entanto, uma regra da prática estabelece que o teste do Qui-quadrado de Pearson não deve ser aplicado se o grau de liberdade³ for igual a 1 e, quando maior que 1, mais de 20% das freqüências esperadas forem menores que 5 ou alguma delas for menor que 1. Apesar do teste do Qui-quadrado de Pearson poder ser aplicado nestas condições, representando um risco calculado, uma maneira de obter maiores freqüências esperadas é combinar categorias relacionadas em novas categorias que, por sua vez, devem ser significativas. Com este intuito, os dados da pesquisa foram recodificados. As 6 categorias presentes no questionário (*Sem Importância*, *Pouco Importante*, *Importante*, *Muito Importante*, *Imprescindível* e *Não Avaliado*) transformaram-se em 3 categorias a serem utilizadas no teste das hipóteses (*Pouco Importante*, *Importante* e *Muito Importante*). Para tal, as categorias *Sem Importância* e *Pouco Importante* do questionário foram combinadas em uma única categoria (*Pouco Importante*). O mesmo foi feito com as categorias *Muito Importante* e *Imprescindível* do questionário, dando

¹ Testes não paramétricos, ao contrário dos paramétricos, impõem menos condições em termos da distribuição dos dados e da escala utilizada para medição dos mesmos, sendo mais adequados para o contexto da pesquisa, baseada em escala nominal e com distribuição dos dados desconhecida.

² A distribuição de freqüência pode ser observada através da construção de uma tabela de freqüência, que tabula cada valor com o número de vezes que ele ocorre em um conjunto de valores. A freqüência relativa é calculada dividindo-se cada freqüência pelo número total de valores.

³ O grau de liberdade é dado por $(r-1)*(k-1)$, onde r é o número de categorias e k é o número de grupos.

origem a uma única categoria, denominada *Muito Importante*. Além disso, as ocorrências da categoria *Não Avaliado* foram substituídas pela categoria que representa não resposta fornecida pelo próprio *software* de Estatística. A combinação de categorias, aceita estatisticamente quando o objetivo é reforçar o teste de hipótese, de fato melhorou a representatividade do estudo, possibilitando a verificação, com 95% de confiança, da independência da importância atribuída a determinados tipos de conhecimento, alcançando, em vários casos, 99% de confiança.

O anexo III apresenta os valores obtidos no teste das hipóteses e a análise descritiva das respostas obtidas para as 96 questões referentes à importância dos vários tipos de conhecimento.

Os tipos de conhecimento livremente acrescentados pelos participantes ao fim do questionário não foram considerados na análise dos dados da pesquisa, pois apenas 25 participantes fizeram acréscimos, havendo grande variação nos tipos de conhecimento acrescentados. Esta parte do questionário cumpriu, no entanto, a sua finalidade, que foi tratar a ameaça à validade da pesquisa representada pela busca de um resultado específico, permitindo que os participantes sugerissem tipos de conhecimento não previstos.

6.6 Interpretação dos Resultados da Pesquisa

Os acrônimos definidos na tabela 6.6 representam grupos de conhecimento e são utilizados nas demais tabelas desta seção para indicar que um tipo de conhecimento pertence a um determinado grupo de conhecimento, simplificando a sua descrição. Além disso, o acrônimo ES substitui o termo Engenharia de Software.

Tabela 6.6 – Acrônimos

Acrônimo	Significado	Acrônimo	Significado
DM	Domínio da Aplicação	DH	Dados Históricos
ORG	Organização	PT	Parceiros Técnicos
MP	Melhores Práticas	FN	Fornecedores
LA	Lições Aprendidas	CL	Clientes
OC	Outros Conhecimentos	LT	Literatura Técnica

Sendo assim, as tabelas 6.7 a 6.10 apresentam, para cada grupo de participantes da pesquisa, os resultados do teste de hipótese referentes a H1, H2 e H3. As tabelas fornecem os fatores dos quais a importância atribuída a cada tipo de conhecimento independe, ou seja, as hipóteses confirmadas, e a importância atribuída ao mesmo. Os testes estatísticos realizados não permitem afirmações quanto aos fatores de dependência.

Tabela 6.7 – Programadores: Independência e Importância dos Tipos de Conhecimento

Código	Grupo	Tipo de Conhecimento	Fatores de Independência	Grau de Importância
5.01.07	ORG	Estrutura Organizacional	Tamanho dos Projetos	Muito Importante
5.01.09	ORG	Distribuição de Competências	Formação e Experiência dos Profissionais	Muito Importante
6.01.01	OC	Tipos de Software	Tamanho das Empresas	Muito Importante
6.02	OC	Roteiros de Documentos	Tamanho das Empresas	Muito Importante
6.04.06	MP	Documentação de Software	Tamanho das Empresas	Muito Importante
6.05.13	LA	Manutenção de Software	Tamanho das Empresas	Muito Importante
6.06.01	DH	Gerência de Projeto	Tamanho dos Projetos	Muito Importante
6.06.02	DH	Gerência de Riscos	Formação e Experiência dos Profissionais	Muito Importante
8.01.02	PT	Restrições, Deficiências e Potenciais	Tamanho dos Projetos	Muito Importante
8.01.04	PT	Produtos e/ou Serviços	Tamanho dos Projetos	Muito Importante
9.01.02	CL	Restrições, Deficiências e Potenciais	Tamanho das Empresas	Muito Importante
9.01.03	CL	Diretrizes e Normas	Tamanho das Empresas	Muito Importante
9.01.04	CL	Produtos e/ou Serviços	Tamanho das Empresas	Importante

Tabela 6.8 – Analistas: Independência e Importância dos Tipos de Conhecimento

Código	Grupo	Tipo de Conhecimento	Fatores de Independência	Grau de Importância
5.01.02	ORG	Restrições, Deficiências e Potenciais	Tamanho das Empresas	Muito Importante
5.01.06	ORG	Produtos e/ou Serviços	Tamanho das Empresas	Muito Importante
5.01.11	ORG	Processos Organizacionais	Tamanho das Empresas	Muito Importante
6.04.01	MP	Modelagem de Processo	Tamanho das Empresas	Muito Importante
6.04.14	MP	Engenharia Reversa e Reengenharia de Software	Tamanho dos Projetos	Muito Importante
6.04.16	MP	Avaliação e Melhoria de Processo de Software	Tamanho dos Projetos	Muito Importante
6.04.20	MP	Processo de Solução de Problemas	Tamanho dos Projetos	Muito Importante
6.05.02	LA	Gerência de Projeto	Tamanho das Empresas	Muito Importante
6.05.03	LA	Gerência de Riscos	Tamanho dos Projetos	Muito Importante
6.05.04	LA	Gerência da Qualidade	Tamanho dos Projetos	Muito Importante
6.05.05	LA	Gerência de Configuração	Formação e Experiência dos Profissionais	Muito Importante
6.05.16	LA	Avaliação e Melhoria do Processo de Software	Tamanho das Empresas	Muito Importante
6.06.01	DH	Gerência de Projeto	Tamanho das Empresas	Muito Importante
6.06.03	DH	Gerência de Qualidade	Tamanho das Empresas	Muito Importante
7.01.02	FN	Restrições, Deficiências e Potenciais	Tamanho dos Projetos	Importante
8.01.04	PT	Produtos e/ou Serviços	Formação e Experiência dos Profissionais	Importante
8.01.06	PT	Alocação dos Profissionais	Formação e Experiência dos Profissionais	Importante
9.01.03	CL	Diretrizes e Normas	Tamanho dos Projetos	Muito Importante
9.01.06	CL	Alocação de Profissionais	Tamanho dos Projetos	Muito Importante
9.01.09	CL	Processos Organizacionais	Tamanho das Empresas e dos Projetos	Muito Importante
10.01.01	LT	Conhecimento Teórico em ES	Tamanho das Empresas	Muito Importante
10.01.02	LT	Inovações Tecnológicas em ES	Tamanho das Empresas	Muito Importante
10.01.04	LT	LA pela Indústria de Software	Tamanho das Empresas	Muito Importante

Tabela 6.9 – Gerentes: Independência e Importância dos Tipos de Conhecimento




Código	Grupo	Tipo de Conhecimento	Fatores de Independência	Grau de Importância
5.01.02	ORG	Restrições, Deficiências e Potenciais	Tamanho dos Projetos	Muito Importante
5.01.04	ORG	Objetivos e Metas Organizacionais	Tamanho dos Projetos	Muito Importante
5.01.07	ORG	Estrutura Organizacional	Formação e Experiência dos Profissionais	Muito Importante
5.01.08	ORG	Alocação dos Profissionais	Tamanho dos Projetos	de Importante a Muito Importante
5.01.09	ORG	Distribuição de Competências	Tamanho dos Projetos	Muito Importante
5.01.12	ORG	Contratos/Acordos	Tamanho dos Projetos	Muito Importante
6.04.04	MP	Gerência de Qualidade	Tamanho dos Projetos	Muito Importante
6.04.06	MP	Documentação de Software	Tamanho dos Projetos	Muito Importante
6.04.07	MP	Engenharia de Sistemas	Tamanho dos Projetos	Muito Importante
6.04.11	MP	Testes	Tamanho dos Projetos	Muito Importante
6.04.15	MP	Reuso de Itens de Software	Tamanho das Empresas	Muito Importante
6.05.03	LA	Gerência de Riscos	Tamanho dos Projetos	Muito Importante
6.05.05	LA	Gerência de Configuração	Tamanho dos Projetos	Muito Importante
6.05.06	LA	Documentação de Software	Tamanho dos Projetos	Muito Importante
6.05.15	LA	Reuso de Itens de Software	Tamanho das Empresas	Muito Importante
7.01.08	FN	Relações Informais	Tamanho das Empresas	Pouco Importante
8.01.04	PT	Produtos e/ou Serviços	Formação e Experiência dos Profissionais	Importante
8.01.07	PT	Distribuição de Competências	Formação e Experiência dos Profissionais	Importante
9.01.05	CL	Estrutura Organizacional	Formação e Experiência dos Profissionais	Muito Importante
9.01.07	CL	Distribuição de Competências	Tamanho dos Projetos	Muito Importante

Tabela 6.10 – Diretores: Independência e Importância dos Tipos de Conhecimento

Código	Grupo	Tipo de Conhecimento	Fatores de Independência	Grau de Importância
5.01.07	ORG	Estrutura Organizacional	Tamanho dos Projetos	de Pouco Importante a Importante
5.01.11	ORG	Processos Organizacionais	Formação e Experiência dos Profissionais	Muito Importante
6.04.05	MP	Gerência de Configuração	Formação e Experiência dos Profissionais	Muito Importante
6.04.19	MP	Processo de Fornecimento de Produto de Software	Tamanho dos Projetos	Muito Importante
6.05.03	LA	Gerência de Riscos	Formação e Experiência dos Profissionais e Tamanho dos Projetos	de Importante a Muito Importante
6.05.08	LA	Análise e Especificação de Requisitos de Software	Formação e Experiência dos Profissionais	Muito Importante
6.05.14	LA	Engenharia Reversa e Reengenharia de Software	Formação e Experiência dos Profissionais, Tamanho das Empresas e dos Projetos	Muito Importante
6.05.15	LA	Reuso de Itens de Software	Formação e Experiência dos Profissionais	Muito Importante
6.05.16	LA	Avaliação e Melhoria do Processo de Software	Tamanho das Empresas	Muito Importante
6.06.05	DH	Métricas de Produto	Formação e Experiência dos Profissionais	Muito Importante
7.01.04	FN	Produtos e/ou Serviços	Formação e Experiência dos Profissionais	Muito Importante
7.01.05	FN	Estrutura Organizacional	Tamanho dos Projetos	Pouco Importante e Muito Importante
8.01.04	PT	Produtos e/ou Serviços	Formação e Experiência dos Profissionais	Muito Importante
9.01.01	CL	Missão das Empresas	Tamanho dos Projetos	Muito Importante
9.01.08	CL	Relações Informais	Tamanho das Empresas	Muito Importante

Na tabela 6.10, o conhecimento sobre a Estrutura Organizacional dos Fornecedores foi considerado tanto *Pouco Importante* quanto *Muito Importante*, que são categorias opostas na escala. A participação de um maior número de diretores, apenas 10 participaram da pesquisa, permitiria melhor identificação da importância atribuída a este tipo de conhecimento.

O resultado do teste das hipóteses, apesar de estatisticamente significativo, não possui, sozinho, importância prática para o estabelecimento de uma estratégia evolutiva para incorporação de conteúdo e desenvolvimento de serviços e ferramentas para uso nos ADSOrg. Desta forma, os 96 tipos de conhecimento mencionados no questionário da pesquisa também foram ordenados de acordo com a importância atribuída pelos participantes da pesquisa, considerando os mesmos grupos de empresas, projetos e participantes anteriormente mencionados, o que significa que os tipos de conhecimento foram ordenados separadamente para cada um dos grupos. Todas as possíveis categorias de importância foram consideradas (*Sem Importância, Pouco Importante, Importante, Muito Importante, Imprescindível e Não Avaliado*). Para a ordenação, foi realizada a análise das frequências dos valores de importância atribuídos aos tipos de conhecimento. Quando procurando pelos tipos de conhecimento considerados mais importantes, foram observados a frequência relativa do valor *Imprescindível*, a frequência relativa acumulada dos valores *Imprescindível* e *Muito importante* e o comportamento da distribuição das frequências. De forma similar, para obter os tipos de conhecimento considerados menos importantes, foram observados a frequência relativa do valor *Sem Importância*, a frequência relativa acumulada dos valores *Sem Importância* e *Pouco Importante*, além do comportamento da distribuição das frequências. Os 25 tipos de conhecimento considerados mais importantes e os 25 tipos de conhecimento considerados menos importantes para cada grupo são apresentados no anexo IV.

Além disso, a fim de obter um resultado mais genérico, ou seja, menos dependente das características específicas de cada grupo, os 96 tipos de conhecimento também foram ordenados de acordo com a sua ocorrência nas tabelas apresentadas no anexo IV, ou seja, de acordo com a sua ocorrência entre os 25 mais importantes e os 25 menos importantes identificados para cada grupo. O raciocínio utilizado considerou que um tipo de conhecimento é mais importante quanto maior for o número de ocorrências do mesmo entre os 25 mais importantes e menor for o número de ocorrências entre os 25 menos importantes. O resultado é apresentado na tabela 6.11, que utiliza cores para auxiliar a discussão, as quais são:  conhecimento relacionado ao domínio de aplicação (DM),  conhecimento sobre a organização que desenvolve e mantém software (ORG),  melhores práticas da organização no

desenvolvimento e na manutenção de software (MP), □ dados históricos da organização relacionados ao desenvolvimento e à manutenção de software (DH), ■ outros conhecimentos relacionados ao desenvolvimento e à manutenção de software (OC), □ conhecimento sobre fornecedores de produtos e/ou serviços (FN), ■ conhecimento sobre parceiros técnicos (PT), □ conhecimento sobre clientes (CL), □ indicação de referências da literatura técnica (LT).

Tabela 6.11 – Classificação Geral dos Tipos de Conhecimentos quanto à Importância atribuída nos Ambientes para Desenvolvimento e Manutenção de Software

	Código	Grupo	Tipo de Conhecimento
1º	6.04.10	MP	Codificação
2º	6.04.11	MP	Testes
3º	6.04.06	MP	Documentação de Software
4º	6.04.04	MP	Gerência de Qualidade
5º	6.01.02	OC	Processo de Software da Organização
5º	6.04.20	MP	Processo de Solução de Problemas
7º	6.04.08	MP	Análise e Especificação de Requisitos de Software
8º	6.04.09	MP	Projeto (<i>design</i>) de Software
9º	4.01	DM	Domínio da Aplicação
10º	6.04.02	MP	Gerência de Projeto
11º	6.04.07	MP	Engenharia de Sistemas
12º	6.02	OC	Roteiros de Documentos com Exemplos de Uso
13º	6.04.16	MP	Avaliação e Melhoria de Processo de Software
14º	6.03	OC	Itens de Software Úteis para Desenvolvimentos Futuros
15º	6.04.01	MP	Modelagem de Processo
16º	6.04.13	MP	Manutenção de Software
17º	6.01.03	OC	Avaliação e Áreas para Melhoria do Processo de Software da Organização
18º	5.01.09	ORG	Distribuição de Competências entre os Profissionais
19º	10.01.03	LT	Relatos sobre as MP da Indústria de Software
20º	5.01.04	ORG	Objetivos e Metas Organizacionais
20º	6.04.17	MP	Treinamento
22º	6.06.01	DH	Gerência de Projeto
23º	6.04.12	MP	Operação de Sistema e Suporte a Usuários
23º	6.01.01	OC	Tipos de Software desenvolvidos na Organização
25º	6.04.15	MP	Reuso de Itens de Software
26º	10.01.02	LT	Indicação de Referências sobre Inovações Tecnológicas em ES
26º	10.01.04	LT	Relatos sobre as LA da Indústria de Software
28º	5.01.05	ORG	Áreas Críticas para o Alcance dos Objetivos Organizacionais
29º	6.06.05	DH	Métricas de Produto
30º	10.01.01	LT	Indicação de Referências sobre Conhecimento Teórico em ES
31º	5.01.02	ORG	Restrições, Deficiências e Potenciais
32º	6.06.03	DH	Gerência de Qualidade
33º	6.06.06	DH	Métricas de Processo
34º	5.01.11	ORG	Processos Organizacionais
35º	5.01.01	ORG	Missão da Organização
36º	6.04.05	MP	Gerência de Configuração
37º	6.04.03	MP	Gerência de Riscos
38º	5.01.03	ORG	Diretrizes e Normas
39º	6.07	OC	Respostas para as Perguntas mais Frequentes entre os Desenvolvedores
40º	5.01.07	ORG	Estrutura Organizacional
41º	9.01.09	CL	Processos Organizacionais
42º	6.04.19	MP	Processo de Fornecimento de Produto de Software
43º	9.01.05	CL	Estrutura Organizacional
44º	9.01.06	CL	Alocação de Profissionais à Estrutura Organizacional
45º	8.01.02	PT	Restrições, Deficiências e Potenciais
46º	5.01.10	ORG	Relações Informais existentes entre os Profissionais
47º	9.01.03	CL	Diretrizes e Normas
48º	9.01.01	CL	Missão da Organização
49º	5.01.12	ORG	Contratos/Acordos com Fornecedores, Parceiros Técnicos e Clientes

Tabela 6.11 – Classificação Geral dos Tipos de Conhecimentos quanto à Importância atribuída nos Ambientes para Desenvolvimento e Manutenção de Software (cont.)

	Código	Grupo	Tipo de Conhecimento
50º	6.06.02	DH	Gerência de Riscos
51º	6.06.04	DH	Treinamento
52º	9.01.02	CL	Restrições, Deficiências e Potenciais
53º	8.01.04	PT	Produtos e/ou Serviços
54º	8.01.03	PT	Diretrizes e Normas
55º	6.04.14	MP	Engenharia Reversa e Reengenharia de Software
56º	6.04.18	MP	Processo de Aquisição de Produto de Software
57º	5.01.06	ORG	Produtos e/ou Serviços
58º	9.01.08	CL	Relações Informais existentes entre os Profissionais
59º	8.01.07	PT	Distribuição de Competências entre os Profissionais
60º	8.01.01	PT	Missão da Organização
61º	7.01.02	FN	Restrições, Deficiências e Potenciais
62º	9.01.04	CL	Produtos e/ou Serviços
63º	5.01.08	ORG	Alocação dos Profissionais à Estrutura Organizacional
64º	9.01.07	CL	Distribuição de Competências entre os Profissionais
65º	8.01.06	PT	Alocação dos Profissionais à Estrutura Organizacional
66º	8.01.08	PT	Relações Informais existentes entre os Profissionais
67º	8.01.05	PT	Estrutura Organizacional
67º	8.01.09	PT	Processos Organizacionais
69º	7.01.04	FN	Produtos e/ou Serviços
70º	7.01.06	FN	Alocação dos Profissionais à Estrutura Organizacional
71º	7.01.01	FN	Missão da Organização
71º	7.01.07	FN	Distribuição de Competências entre os Profissionais
73º	7.01.09	FN	Processos Organizacionais
74º	7.01.03	FN	Diretrizes e Normas
75º	7.01.05	FN	Estrutura Organizacional
75º	7.01.08	FN	Relações Informais existentes entre os Profissionais

Como pode ser observado, a tabela 6.11 não inclui os tipos de conhecimento relacionados às lições aprendidas pela organização no desenvolvimento e na manutenção de software (LA). Três foram as razões para a adoção desta medida: 1) o registro de lições aprendidas é um passo necessário para a definição das melhores práticas da organização, então a importância atribuída às melhores práticas é suficiente para determinar a importância do registro das correspondentes lições aprendidas; 2) a ordem de ocorrência dos tipos de conhecimento referentes às melhores práticas mostrou-se similar à ordem de ocorrência dos tipos de conhecimento referentes às lições aprendidas, sendo que as lições aprendidas foram consideradas, em geral, menos importantes que as melhores práticas; 3) o menor número de tipos de conhecimento facilita a análise da tabela e a formulação de conclusões. Esta medida significa, principalmente, que o questionário aplicado pode ser simplificado, de forma a reunir, em uma única pergunta, as perguntas sobre a importância das melhores práticas e das lições aprendidas referentes a um mesmo aspecto do desenvolvimento e/ou da manutenção de software.

No entanto, cabe destacar, com relação à ordem de ocorrência das lições aprendidas, que: 1) as lições aprendidas sobre análise e especificação de requisitos (6.05.08) aparecem logo após as lições aprendidas sobre codificação (6.05.10), 2) as

lições aprendidas sobre projeto (*design*) (6.05.09) aparecem logo após as lições aprendidas sobre documentação (6.05.06), 3) as lições aprendidas sobre engenharia de sistemas (6.05.07) foram 7 posições mais mal avaliadas que as respectivas melhores práticas, 4) as lições aprendidas sobre operação de sistema e suporte a usuário (6.05.12) foram 4 posições mais bem avaliadas que as respectivas melhores práticas e 5) invertem as posições das respectivas melhores práticas as lições aprendidas sobre avaliação e melhoria de processo (6.05.16) com as lições aprendidas sobre modelagem de processo (6.05.01), as lições aprendidas sobre gerência de configuração (6.05.05) com as lições aprendidas sobre gerência de riscos (6.05.03), e as lições aprendidas sobre engenharia reversa e reengenharia (6.05.14) com as lições aprendidas sobre o processo de aquisição de produto de software (6.05.18).

Para obter o código das melhores práticas correspondentes às lições aprendidas mencionadas e localizá-las na tabela 6.11, basta substituir por 04 os dígitos 05 do código que identifica cada lição aprendida.

6.7 Discussão sobre o Resultado da Pesquisa

As taxas de não avaliação fornecidas no anexo III permitem concluir que tanto analistas quanto programadores tiveram dificuldade em atribuir graus de importância para os tipos de conhecimento referentes aos processos de aquisição e fornecimento de produtos de software (6.04.18, 6.04.19, 6.05.18, 6.05.19) e referentes às lições aprendidas sobre Engenharia Reversa e Reengenharia de Software (6.05.14). Além disso, os programadores também tiveram dificuldade em atribuir graus de importância para contratos/acordos com fornecedores, parceiros técnicos e clientes (5.01.12), para tipos de conhecimento envolvendo Modelagem de Processo, Gerência de Projeto, Gerência de Riscos, Gerência da Qualidade, Gerência de Configuração, Engenharia de Sistemas (6.05.01, 6.05.02, 6.04.03 a 6.04.05, 6.05.03 a 6.05.05, 6.04.07, 6.05.07) e para tipos de conhecimento relacionados a dados históricos (6.06.01 a 6.06.06). Estas dificuldades eram esperadas, uma vez que tais tipos de conhecimento, em geral, não fazem parte do escopo das atividades executadas por esses desenvolvedores e, no caso dos programadores que ainda estão na graduação, os conceitos envolvidos podem ser mesmo desconhecidos. A categoria *Não Avaliado* foi concebida exatamente para atender a esta situação.

Além disso, observando-se a distribuição dos tipos de conhecimento ao longo da tabela 6.11, é possível identificar uma ordem de importância no nível dos grupos de conhecimento contemplados na pesquisa. Tem-se, então, em ordem decrescente de importância: os conhecimentos sobre as melhores práticas da organização e outros conhecimentos relacionados a sua experiência em desenvolvimento e manutenção de software (□■), os conhecimentos relacionados ao domínio da aplicação (■), os conhecimentos referentes à organização que desenvolve e mantém software, conhecimentos provenientes da literatura técnica e conhecimentos que representam dados históricos (□□□), e os conhecimentos referentes aos clientes (□), aos parceiros técnicos (■) e aos fornecedores de produtos e/ou serviços (□) da organização.

Algumas observações ainda podem ser feitas a partir da análise da tabela 6.11:

- Como esperado, os tipos de conhecimento considerados mais importantes, em sua maioria, referem-se à engenharia do produto de software;
- Os participantes, em geral, não atribuíram muita importância ao conhecimento sobre os processos organizacionais. Esperava-se que, em organizações que desenvolvem software para uso próprio, o conhecimento sobre os processos organizacionais (5.01.11) fosse considerado importante, assim como o conhecimento sobre os processos organizacionais das empresas clientes (9.01.09) fosse considerado importante para as organizações que desenvolvem software sob encomenda e/ou para comercialização. Possíveis justificativas residem no fato de apenas uma das organizações envolvidas na pesquisa desenvolver software para uso próprio e do tipo de conhecimento em questão poder ser obtido no contexto do projeto, através da modelagem do negócio ou mesmo da especificação de requisitos. No entanto, projetos de manutenção podem se beneficiar bastante com a gerência deste tipo de conhecimento, principalmente em organizações com alta rotatividade de pessoal, assim como podem também ser beneficiados os novos projetos de desenvolvimento que envolvam processos organizacionais previamente estudados. Além disso, deve-se destacar que o conhecimento sobre os processos organizacionais foi considerado o mais importante dentre os conhecimentos referentes às empresas clientes;
- Conhecimento sobre a estrutura organizacional (5.01.07), sobre a alocação dos profissionais à estrutura organizacional (5.01.08) e sobre a distribuição de competências entre os profissionais da organização (5.01.09) estão relacionados, permitindo a localização das pessoas mais adequadas à solução de um problema ou à execução de uma atividade. No entanto, enquanto o último tipo de conhecimento foi considerado o mais importante no que se refere à organização, o

segundo foi considerado o menos importante. A sugestão, neste caso, seria reformular o questionário de forma a tornar a relação existente mais evidente, por exemplo, reunindo o conhecimento sobre a estrutura organizacional e sobre a alocação dos profissionais a esta estrutura em um único tipo de conhecimento. O mesmo poderia ser feito com os respectivos tipos de conhecimento referentes a parceiros técnicos, fornecedores e clientes;

- Vários participantes responderam o questionário da pesquisa com relação a projetos que envolviam reengenharia de um produto de software já existente, mas, mesmo assim, atribuíram pouca importância ao conhecimento sobre as melhores práticas da organização para Engenharia Reversa e Reengenharia de Software (6.04.14). Apesar de ambos os conceitos estarem descritos no questionário, acredita-se que os participantes não estavam familiarizados com os mesmos, pois também demonstraram dificuldade em classificar seus projetos como sendo de Reengenharia;
- Os participantes, em geral, não atribuíram muita importância aos conhecimentos referentes a Gerência de Riscos (6.04.03 e 6.06.02). Analisando-se as tabelas do anexo IV, verifica-se que, apesar de terem sido considerados importantes por grupos de analistas e de gerentes, estes conhecimentos não foram considerados importantes por vários grupos de diretores, afetando a sua classificação geral;
- O conhecimento referente a parceiros técnicos e fornecedores que podem ser úteis para os desenvolvedores e gerentes de projeto é bem específico. Sobre os parceiros, pode ser importante conhecer restrições impostas, deficiências e potenciais (8.01.02), e, eventualmente, os produtos e/ou serviços oferecidos (8.01.04) e as diretrizes e normas adotadas (8.01.03). Sobre os fornecedores, apenas o conhecimento sobre as restrições impostas, deficiências e potenciais das empresas parece ter alguma importância.

6.8 Conclusão

As tabelas 6.7 a 6.11 e as tabelas apresentadas no anexo IV têm importância prática tanto para o esforço de construção de ADSOrg quanto para qualquer organização que deseje introduzir Gerência do Conhecimento no contexto do desenvolvimento e da manutenção de software. Além disso, outros testes de hipóteses, sobre o mesmo conjunto de dados, podem complementar o estudo feito e a pesquisa pode ser repetida em outras cidades do Brasil e do mundo, ampliando a validade dos resultados obtidos.

Capítulo 7 – Construção de Ambientes Orientados a Organização na Estação TABA

7.1 Introdução

Como mencionado no capítulo 5, a construção de um ADS é complexa. Enquanto a meta é obter ambientes adequados às necessidades específicas de cada projeto de software [22,23], o grande desafio da comunidade é encontrar maneiras de construir e integrar ferramentas de forma que elas, ou suas funcionalidades, possam ser facilmente adaptadas para uso em novos contextos [117].

Segundo HARRISON *et al.* [117], isto significa desenvolver e utilizar arquiteturas e outras abordagens para facilitar a adaptação e a integração, além de construir ferramentas que auxiliem o processo de adaptação e integração.

Este capítulo aborda, então, como a Estação TABA, descrita no capítulo 4, foi estendida para contemplar a construção do modelo de ADSOrg apresentado no capítulo 5, considerando os aspectos destacados por ROCHA *et al.* [22], TRAVASSOS [23] e HARRISON *et al.* [117]. O primeiro passo foi a definição de uma estratégia para administrar a complexidade envolvida na construção de ADSOrg e promover a adaptação e integração. Em seguida, as funções, ambientes, requisitos e arquitetura da Estação TABA foram revisados de acordo com essa estratégia; um processo e um conjunto de atividades complementares foram definidos para conduzir a adaptação; e ferramentas de apoio foram desenvolvidas. No final do capítulo, a execução do processo e das atividades propostas é ilustrada no contexto de uma organização, evidenciando o apoio fornecido pelas ferramentas.

7.2 Estratégia para Construção de ADSOrg na Estação TABA

Com relação aos aspectos destacados por ROCHA *et al.* [22], TRAVASSOS [23] e HARRISON *et al.* [117], o Meta-ambiente da Estação TABA já possuía funcionalidades para auxiliar o engenheiro de software a especificar e instanciar o ADS mais adequado a um projeto específico, quer ele fosse convencional ou orientado a um domínio, integrando ferramentas internas e/ou externas. A estratégia para construção de ADSOrg aqui proposta estabelece um passo anterior à instanciação, que é a configuração do Meta-ambiente para uma organização, de forma a prover o ambiente ideal para o acúmulo e gerência do conhecimento organizacional relevante para as atividades de desenvolvimento e manutenção de software (Ambiente

Configurado), a partir do qual os ADSOrg Instanciados mais adequados aos projetos específicos da organização podem ser criados para disponibilizar conhecimentos, auxiliar a execução das atividades e apoiar o aprendizado organizacional a partir dos projetos. O Ambiente Configurado para uma organização e os respectivos ADSOrg Instanciados contemplam, juntos, os requisitos e componentes propostos para um ADSOrg, representando uma solução para implementação do conceito na qual o Repositório da Organização encontra-se no Ambiente Configurado e os Repositórios dos Projetos nos respectivos ADSOrg Instanciados. A figura 7.1 exibe o esquema utilizado para a construção de ADS na Estação TABA, já englobando a estratégia proposta, que é destacada na cor cinza.

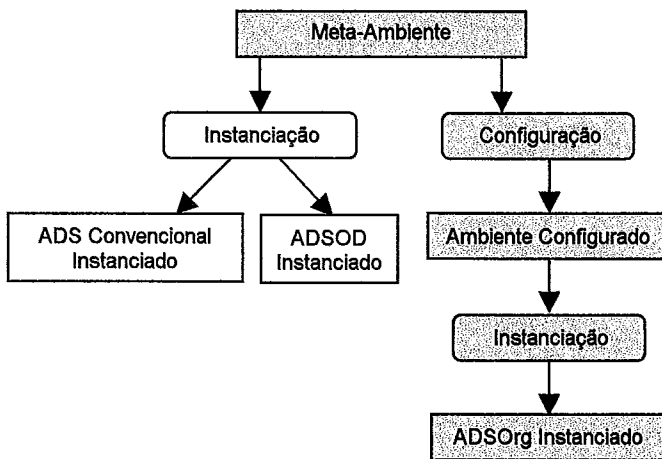


Figura 7.1 – Esquema para Construção de ADS na Estação TABA

7.3 Funções e Ambientes da Estação TABA

Como conseqüência do conceito e tipos de ADSOrg definidos e da estratégia adotada para construção dos mesmos, as funções da Estação TABA e os ambientes que as contemplavam precisaram ser revistos. As funções tornaram-se as seguintes:

- (a) auxiliar o engenheiro de software na configuração do ambiente mais adequado para apoiar o desenvolvimento e a manutenção de software em uma organização específica (Ambiente Configurado), considerando seu processo de software e o conhecimento organizacional relevante neste contexto;
- (b) auxiliar o engenheiro de software na instanciação de ambientes de desenvolvimento de software para projetos específicos (caso a configuração de um ambiente para a organização não seja possível ou considerada adequada);
- (c) auxiliar o engenheiro de software, de empresas cujo negócio é o desenvolvimento e a manutenção de software para diversos clientes, na especialização dos processos da sua empresa de acordo com as particularidades de um determinado cliente;

- (d) auxiliar o engenheiro de software a implementar ferramentas necessárias aos ambientes;
- (e) auxiliar os gerentes de projeto na instanciação de ambientes de desenvolvimento de software para projetos específicos a partir de um Ambiente Configurado;
- (f) apoiar, através dos ADS Instanciados, o desenvolvimento e a manutenção de software, bem como a gerência destas atividades;
- (g) permitir, pelo menos para fins de teste, a execução de produtos de software na própria Estação.

Os ambientes que contemplam essas funções (figura 7.1) podem ser definidos da seguinte maneira:

- **Meta-Ambiente:** ambiente que apóia a configuração de ambientes para organizações específicas, além de apoiar a instanciação de ADS para projetos específicos;
- **Ambiente Configurado:** ambiente configurado a partir do Meta-ambiente que apóia a especialização de processos para clientes (quando pertinente) e a instanciação de ADSOrg para projetos específicos;
- **ADS Instanciado:** ambiente de desenvolvimento de software instanciado a partir de um dos ambientes anteriores, podendo ser convencional, orientado a domínio ou orientado a organização.

7.4 Requisitos dos Ambientes da Estação TABA

Os requisitos da Estação TABA, que foram definidos por TRAVASSOS [23] e estendidos por OLIVEIRA [20] para contemplarem os ADSOD, foram revisados e novamente estendidos, dando origem a um novo conjunto de requisitos, organizados em:

- requisitos gerais,
- requisitos comuns ao Meta-Ambiente e aos Ambientes Configurados,
- requisitos específicos do Meta-Ambiente,
- requisitos específicos dos Ambientes Configurados,
- requisitos adicionais dos Ambientes Configurados para Organizações de Software,
- requisitos comuns a todos ADS Instanciados,
- requisitos comuns a ADSOD e ADSOrg Instanciados,
- requisitos específicos dos ADSOrg Instanciados e
- requisitos adicionais dos ADSOrg Instanciados em Organizações de Software.

➤ Requisitos Gerais

Os requisitos gerais devem ser satisfeitos por qualquer ambiente dentro do contexto da Estação TABA, os quais são:

- ***possuir interface consistente (R1)***: os ambientes da Estação TABA devem possuir mecanismos de interface que permitam a utilização consistente de seus recursos e ferramentas;
- ***possuir um modelo comum de armazenamento de dados (R2)***: a forma de representação das informações nos ambientes da Estação TABA deve possibilitar que as ferramentas compartilhem e utilizem estas informações de forma natural e consistente;
- ***apoiar a reutilização (R3)***: os ambientes da Estação TABA devem fornecer mecanismos que possibilitem a reutilização de qualquer espécie de item de software (código ou não) em contexto diferente do contexto para o qual foi criado;
- ***apoiar o controle de versões e a gerência de configuração (R4)***: os ambientes da Estação TABA devem controlar as modificações feitas nos componentes de conhecimento e/ou itens de software, mantendo-os disponíveis em suas diferentes versões e gerenciando onde essas versões estão sendo utilizadas;
- ***permitir a execução de produtos de software (R5)***: os ambientes da Estação TABA devem possuir funcionalidades que permitam, pelo menos para fins de teste, a execução dos produtos de software neles gerados ou desenvolvidos;

➤ Requisitos Comuns ao Meta-Ambiente e aos Ambientes Configurados

Uma vez que a instanciação de ADS deve ser possível tanto a partir do Meta-Ambiente quanto a partir dos Ambientes Configurados, estes ambientes possuem alguns requisitos em comum. No entanto, nem todos requisitos relacionados à instanciação de ADS são comuns, pois os ambientes instanciados possuem características diferentes. Os requisitos comuns são:

- ***possuir conhecimento sobre processo de software e abordagens de desenvolvimento (R6)***: o Meta-ambiente e os Ambientes Configurados devem possuir conhecimento sobre processo de software e as várias alternativas de modelos de ciclo de vida, paradigmas de desenvolvimento e métodos possíveis de serem utilizados, bem como sobre a adequabilidade da aplicação de cada uma dessas alternativas em diferentes contextos;
- ***apoiar a definição de arquiteturas de referência (R7)***: o Meta-ambiente e os Ambientes Configurados devem apoiar a definição de arquiteturas de software que

servam de referência no desenvolvimento de diferentes produtos de um mesmo tipo e/ou domínio de aplicação;

- ***possuir mecanismo de integração de ferramentas (R8)***: o Meta-ambiente e os Ambientes Configurados devem permitir e facilitar a integração de ferramentas, sejam elas ferramentas internas (desenvolvidas como parte da Estação TABA) ou externas (comerciais ou não).

➤ **Requisitos Específicos do Meta-Ambiente**

Os requisitos específicos do Meta-ambiente da Estação TABA são:

- ***permitir a descrição de tarefas (R9)***: o Meta-ambiente deve possuir mecanismos que facilitem a descrição de tarefas genéricas, que independem de um domínio de aplicação;
- ***apoiar a definição de Teorias de Domínio (R10)***: o Meta-ambiente deve possuir mecanismos que facilitem a definição de Teorias de Domínio para diferentes domínios de aplicação e para o domínio de Engenharia de Software, identificando as tarefas genéricas que são executadas nesses domínios;
- ***apoiar a definição de processos (R11)***: o Meta-ambiente deve apoiar a definição e a especialização do processo padrão de uma organização, além de apoiar a definição de processos para projetos específicos;
- ***apoiar a construção de ferramentas (R12)***: o Meta-ambiente deve possuir funcionalidades que permitam a construção, ou adaptação, de ferramentas a partir de *frameworks* e geradores de ferramentas¹;
- ***gerar Ambientes Configurados para organizações específicas (R13)***: o Meta-ambiente deve ser capaz de gerar um Ambiente Configurado para uma organização específica, considerando os processos padrão e especializados que foram definidos e os domínios de aplicação nos quais a organização atua;
- ***gerar ADS Instanciados para projetos específicos (R14)***: a partir do processo definido para um projeto específico, o Meta-ambiente deve ser capaz de gerar um ADS Instanciado, que pode ser convencional ou orientado a domínio;
- ***permitir a incorporação de novos conhecimentos e experiências (R15)***: o Meta-ambiente deve possuir mecanismos que permitam incorporar conhecimentos e experiências registrados nos ambientes configurados e instanciados como, por exemplo, evoluções em Teorias de Domínio e novas melhores práticas;

¹ Quando uma ferramenta não puder ser construída, ou adaptada, a partir dos *frameworks* e geradores de ferramentas disponíveis no meta-ambiente, um ADS deve ser instanciado com a finalidade de apoiar o desenvolvimento da ferramenta.

- **permitir novas configurações sem perda do conhecimento organizacional (R16):** o Meta-ambiente deve permitir a configuração de um novo ambiente para uma organização (nova versão do Ambiente Configurado), sem acarretar a perda do conhecimento acumulado na versão anterior.

➤ **Requisitos Específicos dos Ambientes Configurados**

Os requisitos que devem ser satisfeitos pelos Ambientes Configurados, quer estes ambientes sejam configurados para Organizações de Software ou para Organizações com outro Tipo de Negócio, são:

- **permitir a evolução das Teorias de Domínio que fazem parte do ambiente (R17):** um Ambiente Configurado deve permitir que novos conceitos e relações, bem como novas instâncias de conceitos e relações, sejam incluídos nas Teorias dos Domínios de aplicação e de Engenharia de Software que fazem parte do ambiente;
- **permitir a descrição da estrutura organizacional (R18):** um Ambiente Configurado deve permitir a descrição da estrutura organizacional e a definição das competências desejadas para cada posição definida pela estrutura;
- **permitir a descrição dos profissionais da organização (R19):** um Ambiente Configurado deve permitir a descrição do perfil dos profissionais da organização e a alocação dos mesmos à estrutura organizacional;
- **permitir a descrição dos processos organizacionais (R20):** um Ambiente Configurado deve permitir a descrição dos processos organizacionais, o que envolve representação gráfica destes processos e descrição dos elementos representados;
- **apoiar a definição de processos para projetos específicos a partir dos processos especializados (R21):** um Ambiente Configurado deve apoiar a definição de um processo para um projeto de software específico, a partir de um dos processos especializados da organização, considerando, para isto, as características do projeto;
- **gerar ADSOrg Instanciados para projetos específicos (R22):** um Ambiente Configurado deve ser capaz de gerar um ADSOrg Instanciado para um projeto de software específico a partir do processo definido para o projeto;
- **apoiar a produção de conhecimento organizacional (R23):** um Ambiente Configurado deve apoiar a filtragem e empacotamento de conhecimento adquirido nos ADSOrg Instanciados e seu armazenamento no repositório da organização, o que inclui a associação do mesmo com as atividades dos processos às quais se refere e com os conceitos e instâncias de conceitos que o descrevem.

➤ **Requisitos Adicionais dos Ambientes Configurados para Organizações de Software**

Ambientes Configurados para Organizações de Software devem satisfazer, em adição aos requisitos mencionados acima, os seguintes requisitos:

- **permitir a descrição da estrutura organizacional dos clientes (R24):** um Ambiente Configurado para uma Organização de Software deve permitir a descrição da estrutura organizacional dos clientes da organização;
- **permitir a descrição dos profissionais das organizações clientes (R25):** um Ambiente Configurado para uma Organização de Software deve permitir a descrição do perfil dos profissionais das organizações clientes e a alocação dos mesmos a suas respectivas estruturas organizacionais;
- **permitir a descrição dos processos organizacionais dos clientes (R26):** um Ambiente Configurado para uma Organização de Software deve permitir a descrição dos processos organizacionais dos clientes da organização;
- **apoiar a especialização de processos de acordo com os clientes (R27):** um Ambiente Configurado para uma Organização de Software deve apoiar, sempre que necessário, a especialização de seus processos de software de acordo com as particularidades de um determinado cliente.

➤ **Requisitos Comuns a Todos os ADS Instanciados**

Independente do ADS Instanciado ser convencional, orientado a domínio ou orientado à organização, os seguintes requisitos devem ser satisfeitos:

- **apoiar a execução do processo e a sua gerência (R28):** um ADS Instanciado deve apoiar a execução do processo e a gerência do mesmo, através de orientação, automação e/ou monitoração do processo e, quando necessário, do suporte a sua modificação;
- **apoiar a execução de atividades (R29):** um ADS Instanciado deve apoiar a execução das atividades do processo, fornecendo acesso às ferramentas associadas a estas atividades;
- **apoiar o trabalho cooperativo (R30):** um ADS Instanciado deve definir protocolos de coordenação, colaboração e comunicação que facilitem o trabalho em equipe, o que é especialmente importante para projetos de desenvolvimento ou manutenção de software em larga escala;
- **possuir suporte para a avaliação do produto e do processo (R31):** um ADS Instanciado deve apoiar a medição dos produtos gerados ao longo do processo, de forma a permitir a garantia da qualidade do produto final e a avaliação do processo.

➤ **Requisitos Comuns a ADSOD e ADSOrg Instanciados**

Tanto ADSOD quanto ADSOrg Instanciados pressupõem que o conhecimento sobre o domínio da aplicação seja utilizado para apoiar as atividades dos desenvolvedores de software. Desta forma, tais ambientes possuem como requisitos comuns:

- **permitir a instanciação da Teoria do Domínio referente à aplicação (R32):** ADSOD e ADSOrg Instanciados devem permitir que novas instâncias de conceitos e relações sejam incluídas na Teoria do Domínio referente à aplicação;
- **fornecer acesso ao conhecimento sobre o domínio da aplicação (R33):** este requisito refere-se à necessidade de ADSOD e ADSOrg Instanciados oferecerem mecanismos e ferramentas de acesso ao conhecimento sobre o domínio da aplicação.

➤ **Requisitos Específicos dos ADSOrg Instanciados**

Os requisitos específicos dos ADSOrg Instanciados são:

- **permitir a instanciação da Teoria do Domínio de Engenharia de Software (R34):** um ADSOrg Instanciado deve permitir que novas instâncias de conceitos e relações sejam incluídas na Teoria do Domínio de Engenharia de Software;
- **fornecer acesso ao conhecimento sobre Engenharia de Software (R35):** este requisito refere-se à necessidade de um ADSOrg Instanciado oferecer mecanismos e ferramentas de acesso ao conhecimento sobre o domínio de Engenharia de Software;
- **apoiar a localização de profissionais (R36):** um ADSOrg Instanciado deve apoiar a localização dos profissionais da organização mais adequados para auxiliar na execução de uma atividade ou na solução de um problema;
- **apoiar o entendimento dos processos organizacionais (R37):** um ADSOrg Instanciado deve permitir a visualização dos processos organizacionais e a navegação através dos seus diferentes níveis de abstração, fornecendo, sob solicitação, detalhes sobre os elementos representados e permitindo acesso às informações e conhecimentos disponíveis no ambiente;
- **fornecer acesso ao conhecimento organizacional (R38):** este requisito refere-se à necessidade do ADSOrg Instanciado oferecer, de acordo com a atividade do processo sendo executada, mecanismos e ferramentas de acesso ao conhecimento acumulado pela organização ao longo do tempo;

- **apoiar a aquisição de conhecimento para a organização (R39):** um ADSOrg deve oferecer mecanismos que permitam a aquisição de conhecimento ao longo das atividades do processo e ao final do projeto.

➤ **Requisitos Adicionais dos ADSOrg Instanciados em Organizações de Software**

ADSOrg Instanciados a partir de Ambientes Configurados para Organizações de Software possuem, além de todos os requisitos definidos para os ADSOrg Instanciados, os seguintes requisitos:

- **permitir a evolução do modelo da organização cliente (R40):** um ADSOrg Instanciado deve permitir que o modelo da organização cliente, já disponível no Ambiente Configurado e composto da estrutura organizacional, dos profissionais alocados a esta estrutura e dos processos organizacionais, seja atualizado e expandido de acordo com as informações obtidas no projeto corrente;
- **apoiar a localização de profissionais da organização cliente (R41):** um ADSOrg Instanciado deve apoiar a localização dos profissionais da organização cliente que sejam mais adequados para auxiliar na execução de uma atividade ou na solução de um problema;
- **apoiar o entendimento dos processos organizacionais do cliente (R42):** um ADSOrg Instanciado deve permitir a visualização dos processos organizacionais do cliente e a navegação através dos seus diferentes níveis de abstração, fornecendo, sob solicitação, detalhes sobre os elementos representados e permitindo acesso às informações e conhecimentos disponíveis no ambiente.

7.5 Processos para Adaptação de Ambientes na Estação TABA

Tanto a configuração do Meta-ambiente para uma organização, originando um Ambiente Configurado, quanto a instanciação de ambientes a partir Meta-ambiente ou do Ambiente Configurado, originando os ADS Instanciados, envolvem a execução de um conjunto de atividades de adaptação que devem ser apoiadas por serviços e ferramentas, tendo sido contempladas nos requisitos definidos na seção anterior. No entanto, para fornecer apoio efetivo aos engenheiros de software e gerentes de projeto, não só um conjunto de serviços e ferramentas é importante, mas também a definição de processos que descrevam as atividades a serem executadas e como elas estão relacionadas. Desta forma, verificou-se a necessidade de definição de dois processos para a implementação da estratégia proposta na seção 7.2: um para

orientar a configuração do Meta-ambiente para uma organização e outro para orientar a instanciação de ADSOrg a partir de um Ambiente Configurado.

O processo de configuração foi definido como parte deste trabalho, sendo apresentado, de forma resumida, na tabela 7.1. A descrição detalhada do processo, que inclui orientações para a execução das atividades, é fornecida no anexo V.

Tabela 7.1 – Processo de Configuração do Meta-ambiente TABA

Atividade 1: Contextualizar Configuração
<p>Objetivo: contextualizar a configuração, indicando as características gerais da organização, a sua cultura na área de software e quais são os objetivos almejados com a configuração, de forma a fornecer os subsídios necessários à elaboração da proposta de configuração.</p> <p>Responsáveis: Configurador de Ambientes e representantes da organização.</p> <p>Sub-atividades:</p> <ul style="list-style-type: none">• <u>Realizar entrevistas</u> para coletar informações necessárias às demais sub-atividades;• <u>Caracterizar a organização</u>, registrando informações que indiquem o seu tamanho e área de atuação;• <u>Identificar cultura organizacional na área de software</u>, o que envolve tipos de software produzidos, tamanho e complexidade dos projetos, tecnologias utilizadas, entre outros;• <u>Identificar objetivos para a configuração do ambiente</u> a partir dos objetivos identificados para a organização na área de software.
Atividade 2: Elaborar Proposta para Configuração do Ambiente
<p>Objetivo: elaborar a proposta de configuração de um ambiente para a organização, o que implica em especificar o conteúdo e as ferramentas que serão disponibilizadas no Ambiente Configurado e em elaborar planos de organização e custos, além de um cronograma para a configuração.</p> <p>Responsável: Configurador de Ambientes.</p> <p>Sub-atividades:</p> <ul style="list-style-type: none">• <u>Identificar as Teorias de Domínio</u> que serão disponibilizadas no Ambiente Configurado, definindo os seus respectivos propósitos;• <u>Identificar os processos</u> a serem definidos e/ou que precisam ser revistos, de forma a fornecer para a organização um processo padrão e processos especializados para cada contexto de desenvolvimento de software existente na organização;• <u>Identificar as ferramentas</u> a serem disponibilizadas para uso no Ambiente Configurado e/ou nos Ambientes Instanciados a partir dele;• <u>Definir equipes e responsabilidades</u>, estabelecendo quem será o coordenador do projeto de configuração do Meta-ambiente TABA para a organização;• <u>Definir cronograma</u> para a execução das macro-atividades que compõem a proposta de configuração;• <u>Definir os custos</u> por macro-atividades, estabelecendo também o custo total da configuração do Meta-ambiente TABA para a organização;• <u>Enviar a proposta</u> elaborada para que os Representantes da Organização a analisem e forneçam um parecer sobre a mesma.
Atividade 3: Registrar Parecer sobre a Proposta
<p>Objetivo: registrar o parecer dos Representantes da Organização sobre a proposta de configuração, indicando a sua aprovação, a necessidade de modificações e ajustes, ou a não aprovação da proposta.</p> <p>Responsável: Configurador de Ambientes.</p>

Tabela 7.1 – Processo de Configuração do Meta-ambiente TABA (continuação)

Atividade 4: Definir Processo Padrão

Objetivo: definir o processo padrão da organização, que pode contemplar os processos de desenvolvimento e de manutenção de software.

Responsáveis: Configurador de Ambientes.

Sub-atividades:

- Caracterizar o processo padrão a ser definido, estabelecendo quais são os processos de ciclo de vida que devem ser contemplados e se as definições dos processos de ciclo de vida devem ser baseadas ou não em processos já definidos;
- Incluir as atividades do processo padrão selecionado como base na definição do novo processo padrão (quando pertinente);
- Incluir atividades ISO/IEC 12207 no processo padrão da organização;
- Incluir atividades próprias do desenvolvimento ou da manutenção de determinados tipos de software. Atividades desta natureza só devem ser incluídas no processo padrão de uma organização quando são aplicáveis a qualquer projeto de software conduzido na organização;
- Incluir atividades próprias da organização para a qual o ambiente está sendo configurado;
- Incluir atividades orientadas a domínio no processo padrão da organização;
- Estabelecer as atividades obrigatórias do processo padrão, as quais não podem ser retiradas dos processos que serão posteriormente definidos a partir dele;
- Adaptar, para a organização, os roteiros de documentos disponíveis no Meta-ambiente TABA.

Atividade 5: Definir Processo Especializado

Objetivo: definir, a partir do processo padrão, processos especializados para os paradigmas de desenvolvimento utilizados na organização e, opcionalmente, para diferentes tipos de software.

Responsáveis: Configurador de Ambientes.

Sub-atividades:

- Especializar atividades do processo padrão, de forma a torná-las mais adequadas ao paradigma e respectivos métodos de desenvolvimento;
- Incluir atividades próprias do desenvolvimento ou da manutenção de determinados tipos de software, dando origem a um processo especializado tanto para o paradigma quanto para um ou mais tipos de software;
- Estabelecer as atividades obrigatórias do processo especializado, as quais não podem ser retiradas durante a definição de processos para projetos específicos;
- Especializar roteiros de documentos definidos para a organização, quando necessário e de acordo com as características do processo especializado.

Atividade 6: Definir Teoria do Domínio

Objetivo: definir uma Teoria do Domínio para um determinado domínio de conhecimento.

Responsáveis: Engenheiro de Conhecimento e especialistas no domínio.

Sub-atividades:

- Revisar o propósito estabelecido para a Teoria do Domínio durante a elaboração da Proposta para Configuração do Ambiente;
- Especificar os requisitos da Teoria do Domínio através da formulação de questões de competência;
- Capturar a Teoria do Domínio, ou seja, identificar conceitos, relações, propriedades e restrições da ontologia do domínio que compõe a Teoria do Domínio;

Tabela 7.1 – Processo de Configuração do Meta-ambiente TABA (continuação)

Atividade 6: Definir Teoria do Domínio (continuação)
<ul style="list-style-type: none"> • <u>Conceituar a Teoria do Domínio</u>, ou seja, gerar definições precisas e não ambíguas para a ontologia capturada; • <u>Formalizar a Teoria do Domínio</u>, ou seja, descrever a ontologia do domínio utilizando uma linguagem de representação que elimine possíveis ambigüidades; • <u>Identificar as tarefas</u> que são executadas no contexto definido por cada subteoria do domínio; • <u>Avaliar a Teoria do Domínio</u>, de forma a assegurar que ela cumpre o seu propósito, satisfazendo os requisitos estabelecidos e realmente sendo capaz de descrever o mundo real para o qual foi projetada; • <u>Implementar a Teoria do Domínio</u> no Meta-ambiente, possivelmente integrando subteorias do domínio já disponíveis.
Atividade 7: Descrever Tarefa
<p>Objetivo: descrever uma tarefa genérica que tenha sido identificada como pertinente ao contexto de uma subteoria do domínio.</p> <p>Responsáveis: Engenheiro de Conhecimento e especialistas na execução da tarefa.</p> <p>Sub-atividades:</p> <ul style="list-style-type: none"> • <u>Descrever a tarefa em linguagem natural</u>, o que significa descrever o objetivo da tarefa e os elementos (objetos e ações) que são necessários para a sua solução; • <u>Definir a estratégia para solução da tarefa</u>, o que significa escolher um método para a solução da tarefa e, a depender da estratégia adotada pelo método, descrever a solução do problema em linguagem natural; • <u>Capturar conceitos e relações da tarefa</u> a partir dos objetos mencionados na descrição em linguagem natural da tarefa; • <u>Descrever o fluxo de controle da tarefa</u> em linguagem estruturada, quando a tarefa tiver sido considerada composta; • <u>Formalizar a Descrição da Tarefa</u>, ou seja, descrever em linguagem formal tanto os conceitos, relações e restrições da ontologia da tarefa quanto o fluxo de controle ou regras de inferência anteriormente definidos, eliminando ambigüidades; • <u>Avaliar a Descrição da Tarefa</u>, de forma a assegurar que ela realmente descreve o problema proposto pela tarefa e apresenta uma solução para o problema que seja válida no mundo real; • <u>Implementar a Descrição da Tarefa</u> no Meta-ambiente, integrando, quando a tarefa for composta, as descrições de suas sub-tarefas.
Atividade 8: Incluir Novas Ferramentas nas Descrições dos Processos
<p>Objetivo: incluir a referência, na descrição dos processos padrão e/ou especializados, às ferramentas que tenham sido construídas após a definição dos mesmos, de forma que, no momento da criação do Ambiente Configurado (atividade 9), possam ser integradas ao ambiente.</p> <p>Responsável: Configurador de Ambientes.</p>
Atividade 9: Criar Ambiente Configurado
<p>Objetivo: criar o Ambiente Configurado para a organização.</p> <p>Responsáveis: Configurador de Ambientes.</p> <p>Sub-atividades:</p> <ul style="list-style-type: none"> • <u>Definir o Ambiente Configurado</u>, selecionando o processo padrão e as Teorias de Domínio a serem fornecidas como conteúdo inicial do ambiente; • <u>Gerar o Ambiente Configurado</u> (código executável); • <u>Testar o Ambiente Configurado</u> para assegurar que ele foi gerado corretamente.

Uma vez que o Ambiente Configurado é entregue à organização, o Gerente de Conhecimento prepara o ambiente para utilização. A tabela 7.2 apresenta, de forma resumida, a descrição das atividades executadas com este intuito. Uma descrição mais detalhada, que inclui orientações para execução das atividades, é fornecida no anexo VI.

Tabela 7.2 – Atividades de Preparação para Uso do Ambiente Configurado

Atividade: Descrever Organização
<p>Objetivo: obter e registrar informações sobre a organização, de forma que estas informações possam ser disponibilizadas para apoiar os seus vários projetos de software.</p> <p>Responsáveis: Gerente de Conhecimento e especialistas das áreas envolvidas.</p> <p>Sub-atividades:</p> <ul style="list-style-type: none"> • <u>Descrever Aspectos Gerais</u>, ou seja, obter e registrar no Ambiente Configurado informações que definem como a organização interage com o ambiente externo; • <u>Descrever Estrutura Organizacional</u>, ou seja, obter e registrar no Ambiente Configurado informações sobre como a organização se encontra estruturada; • <u>Descrever Processos Organizacionais</u>, o que significa obter informações sobre os processos executados na organização, representando-os graficamente no Ambiente Configurado;
Atividade: Descrever Organização Cliente
<p>Objetivo: obter e registrar informações sobre uma organização cliente, de forma que estas informações possam ser disponibilizadas para apoiar os vários projetos de software que envolvem esta organização. Esta atividade só deve ser executada se a organização tem como atividade de negócio o desenvolvimento e manutenção de software.</p> <p>Responsáveis: Gerente de Conhecimento e especialistas em projetos com o cliente.</p> <p>Sub-atividades:</p> <ul style="list-style-type: none"> • <u>Descrever Aspectos Gerais</u>, ou seja, obter e registrar no Ambiente Configurado informações que definem como a organização cliente interage com o ambiente externo; • <u>Descrever Estrutura Organizacional</u>, ou seja, obter e registrar no Ambiente Configurado informações sobre como a organização cliente se encontra estruturada; • <u>Descrever Processos Organizacionais</u>, o que significa obter informações sobre os processos executados na organização cliente, representando-os graficamente no Ambiente Configurado;
Atividade: Adquirir Conhecimento Inicial
<p>Objetivo: disponibilizar, no Ambiente Configurado, itens de conhecimento sobre desenvolvimento e manutenção de software que já estavam disponíveis para a organização antes da configuração em algum formato, local e meio de armazenamento. Esta atividade deve ser executada segundo o processo de captura de conhecimento proposto por MONTONI [31].</p> <p>Responsáveis: Gerente de Conhecimento, Comitê de Avaliação e antigos detentores dos conhecimentos.</p>

A descrição tanto da organização quanto de possíveis organizações clientes deve ser feita de forma evolutiva, sempre considerando a relação custo-benefício. Além disso, o conteúdo do ambiente também irá evoluir a cada novo projeto da organização.

Por fim, o processo de instanciação de ADSOrg foi definido por BERGER [30], também como parte do esforço para a construção de ADSOrg na Estação TABA.

7.6 Arquitetura da Estação TABA

A figura 7.2 mostra, em termos de funcionalidades e repositórios, a organização dos ambientes da Estação TABA que implementam a estratégia proposta para construção de ADSOrg, contemplando os requisitos estabelecidos e apoiando os processos de configuração e instanciação.

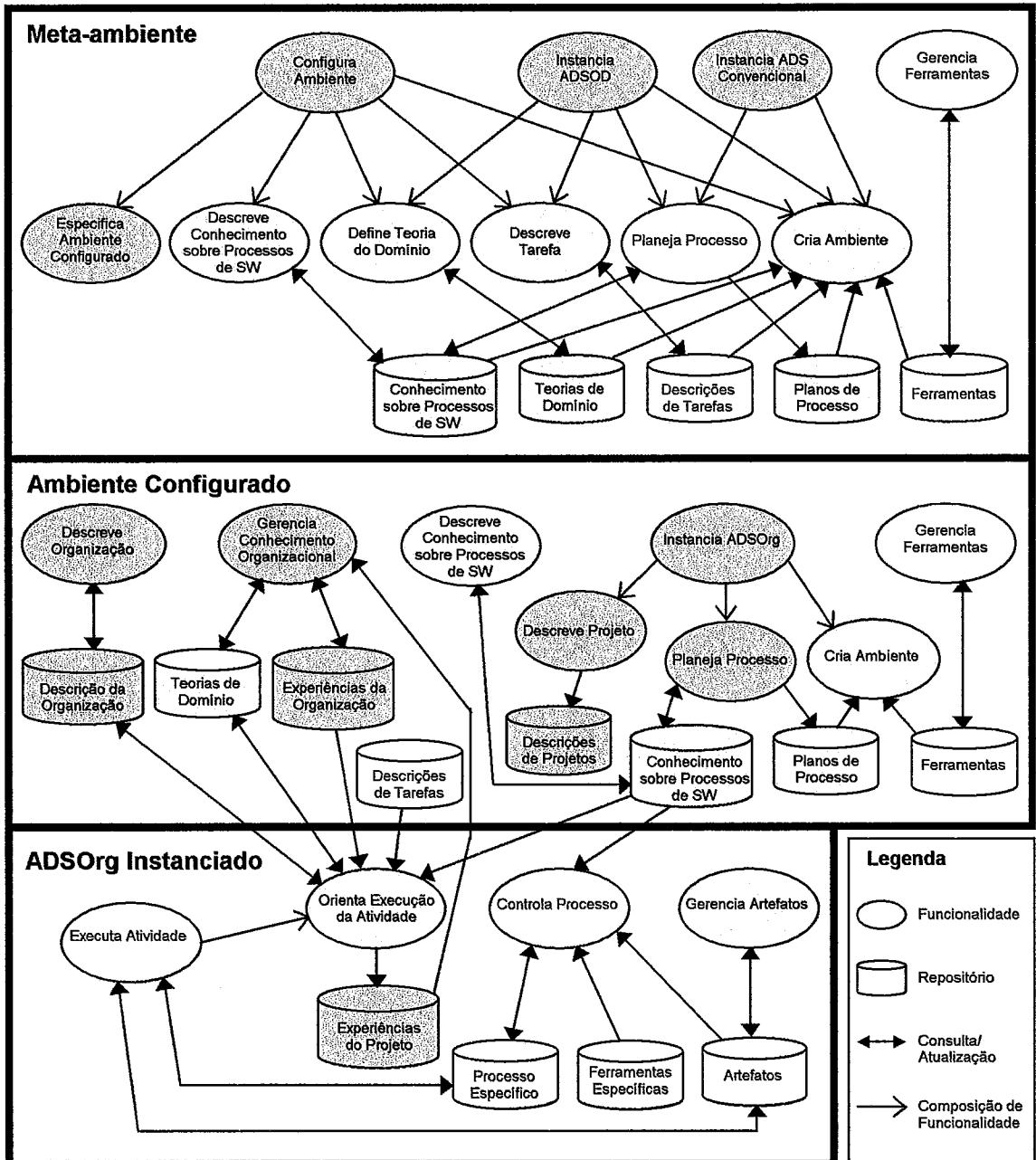


Figura 7.2 – Principais Componentes do Meta-ambiente, Ambientes Configurados e ADSOrg Instanciados

Os componentes em cinza escuro foram acrescentados à Estação TABA; os componentes em cinza claro já existiam, mas foram estendidos ou sofreram alterações; e os componentes em branco não foram estendidos nem alterados. As funcionalidades *Instancia ADSOD* e *Instancia ADS Convencional* foram acrescentadas para que a interação com o usuário seja consistente, ou seja, sempre oriente o usuário em termos das funcionalidades a serem utilizadas para atingir um determinado objetivo.

As figuras 7.3 e 7.4 apresentam a organização dos ADSOD e ADS Instanciados, completando-se, assim, o contexto para discussão sobre as arquiteturas dos ambientes que passaram a constituir a Estação TABA.

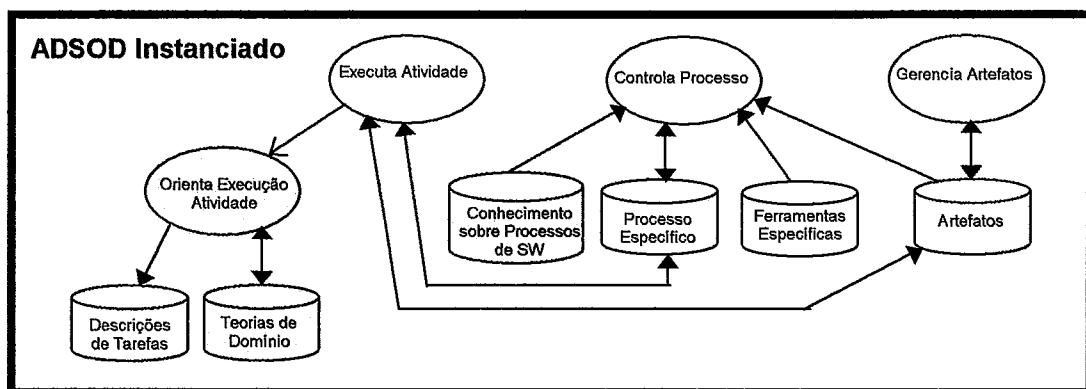


Figura 7.3 – Principais Componentes dos ADSOD Instanciados

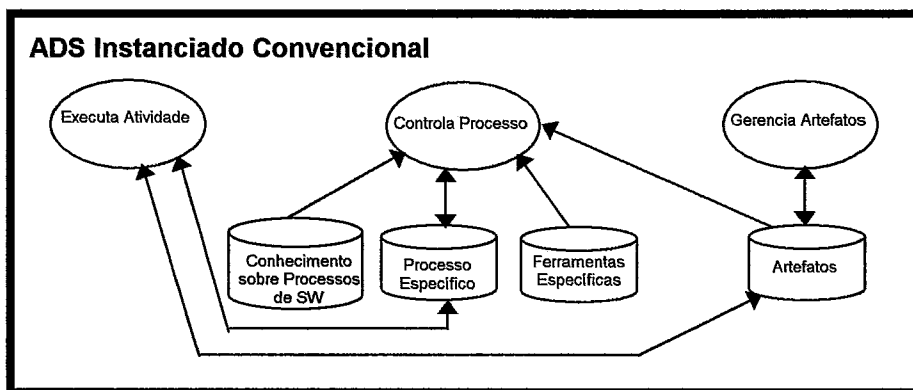


Figura 7.4 – Principais Componentes dos ADS Instanciados Convencionais

Sendo assim, as principais funcionalidades providas pelo *Meta-ambiente* são: apoiar a configuração de ambientes para as organizações, apoiar a instanciação de ADSOD e de ADS convencionais, além de gerenciar ferramentas (figura 7.2). As três primeiras funcionalidades envolvem, como indicado na figura 7.2, alguma combinação das funcionalidades: apoiar a especificação do Ambiente Configurado, apoiar a descrição de conhecimento sobre processos de *software*, apoiar a definição de Teorias de Domínio, apoiar a descrição de tarefas, apoiar o planejamento do processo

para um projeto e apoiar a criação de ambientes. *Apoiar a especificação do Ambiente Configurado* implica em apoiar a identificação das necessidades da organização em relação ao Ambiente Configurado, apoiando também a elaboração de uma proposta de configuração. *Apoiar a descrição de conhecimento sobre processos de software* implica em apoiar tanto a descrição dos elementos que compõem os processos (atividades, métodos, técnicas, entre outros), como proposto por OLIVEIRA [20], quanto a definição dos processos padrão e especializados da organização. Da mesma forma, *apoiar a definição de uma Teoria do Domínio* implica não só em permitir a introdução da Teoria do Domínio no Meta-ambiente [20], mas também apoiar a construção da Teoria do Domínio. O *apoio à descrição de tarefas* também significa apoiar tanto a construção da Descrição da Solução do Problema² referente a uma tarefa quanto a sua introdução no Meta-ambiente. *Apoiar a criação de ambientes* refere-se à geração e teste dos ambientes configurados ou instanciados. A funcionalidade *apoiar o planejamento do processo* é necessária no Meta-ambiente para possibilitar a instanciação de ambientes, orientados a domínio ou não, para projetos específicos. Como já mencionado, a instanciação de ambientes a partir do Meta-ambiente é utilizada quando a configuração de um ambiente para a organização não é possível ou não é considerada adequada. Por não ser importante no contexto deste trabalho, não houve evolução na implementação dessa funcionalidade. Por fim, nenhum acréscimo também foi feito à funcionalidade *gerenciar ferramentas*, que permanece como descrito por OLIVEIRA [20], permitindo o uso de ferramentas internas e externas no Meta-ambiente e nos ambientes configurados e instanciados.

Nos **Ambientes Configurados** (figura 7.2), as funcionalidades básicas são: apoiar a descrição da organização, apoiar a gerência do conhecimento organizacional, apoiar a descrição de conhecimento sobre processos de software, apoiar a instanciação de ADSOrg, além de gerenciar ferramentas. Tanto o *apoio à descrição de conhecimento sobre processos de software* quanto a *gerência de ferramentas* já foram discutidos, cabendo destacar que a presença da primeira funcionalidade nos Ambientes Configurados visa não só permitir pequenas alterações nos processos padrão e especializados da organização, tais como a inclusão de atividades e a alteração dos elementos associados (por exemplo: artefatos e recursos), mas também permitir, quando pertinente, a especialização dos processos da organização para um determinado cliente. No entanto, estes aspectos não foram implementados como parte deste trabalho e serão explorados no contexto de trabalhos futuros destinados à

² Descrição de Solução de Problema é a nomenclatura usada por ZLOT [137] para indicar que a descrição da tarefa contempla tanto a ontologia da tarefa, com a definição dos conceitos envolvidos na solução do problema, quanto o método utilizado para solução do mesmo.

identificação de funcionalidades específicas para Ambientes Configurados e ADSOrg Instanciados para Organizações de Software. *Apoiar a descrição da organização* significa apoiar a descrição de sua estrutura, seus profissionais e processos organizacionais. O *apoio à gerência do conhecimento organizacional* implica no apoio à evolução das Teorias de Domínio disponíveis no ambiente e no apoio à produção de conhecimento organizacional a partir dos conhecimentos adquiridos em projetos específicos. Os nomes dos repositórios são *Experiências da Organização* e *Experiências do Projeto* por conterem conhecimento que foi adquirido através da experiência, que é complementar ao conhecimento teórico. *Apoiar a instanciação de ADSOrg* para projetos específicos significa apoiar a descrição dos projetos, o planejamento dos seus respectivos processos e a criação dos ambientes propriamente ditos. Neste sentido, *apoiar a descrição de um projeto* implica em apoiar a descrição de suas características e das características do produto a ser desenvolvido ou mantido, de forma a possibilitar o *apoio ao planejamento do processo* através da sugestão de modelos de ciclo de vida, de mapeamentos entre atividades do processo e o modelo de ciclo de vida selecionado, e da sugestão das técnicas de avaliação da qualidade mais adequadas ao projeto.

Como pode ser observado nas figuras 7.2, 7.3 e 7.4, todos os **ADS Instanciados** têm como funcionalidades controlar a execução do processo, apoiar a execução das atividades de engenharia e gerência, além de apoiar a gerência de artefatos. *Controlar a execução do processo* implica em fornecer acesso a roteiros, ferramentas e artefatos de acordo com a atividade sendo executada, além de apoiar o acompanhamento e a alteração do processo de software durante a sua execução conforme proposto por ARAÚJO [139] e evoluído por MAFRA [36]. *Apoiar a gerência de artefatos* refere-se ao apoio ao acesso compartilhado dos artefatos, ao controle de versões e à gerência de configuração. A diferença entre os tipos de ADS reside exatamente no *apoio dado à execução das atividades de engenharia e gerência*. Em um **ADS Instanciado Convencional**, este apoio significa apenas a automatização, em algum grau, da execução destas atividades, o que conduz à produção de artefatos, registro de informações sobre a execução da atividade e/ou coleta de informações sobre a execução do processo como um todo. Os **ADSOD Instanciados** já prevêm o fornecimento do conhecimento necessário ao entendimento do domínio para *orientar a execução das atividades*, podendo resultar na evolução da Teoria do Domínio. Por fim, os **ADSOrg Instanciados** expandem a orientação dada à execução das atividades, fornecendo informações e conhecimentos sobre a organização, sobre o domínio de aplicação e de Engenharia de Software, sobre o processo de software, além do conhecimento organizacional em desenvolvimento e manutenção de software

adquirido através da experiência em projetos específicos. Experiências do projeto são coletadas e as Teorias de Domínio e a descrição da organização podem ser evoluídas.

7.7 Considerações Gerais de Implementação

OLIVEIRA [20] elaborou um modelo com as classes conceituais necessárias à re-implementação da Estação TABA em C++, considerando, como mencionado no capítulo 4, as funcionalidades de especificação, instanciação e execução de ambientes. Este modelo baseou-se no modelo definido por TRAVASSOS [23] para a versão da Estação TABA implementada em Eiffel, sendo organizado em 6 pacotes, os quais são: Estação TABA, Ambientes, Conhecimento, Dados, Ferramentas e Controle. Além disso, os trabalhos posteriores ao de OLIVEIRA [20], realizados no contexto da Estação TABA [137,144,149], implicaram na alteração de algumas partes do modelo e na inclusão de novas classes em seus pacotes.

Para a implementação das novas funcionalidades e extensão das funcionalidades já existentes como proposto nas figuras 7.2, 7.3 e 7.4, o modelo da Estação TABA foi revisto, alterado e expandido. As principais alterações foram decorrentes da necessidade de adequação do modelo à representação da ontologia de organização apresentada no capítulo 5.

Parte do modelo de classes da Estação TABA é apresentado no anexo VII e a figura 7.5 exhibe o diagrama principal de pacotes. O conteúdo atual dos pacotes e as alterações que foram necessárias estão descritos na tabela 7.3.

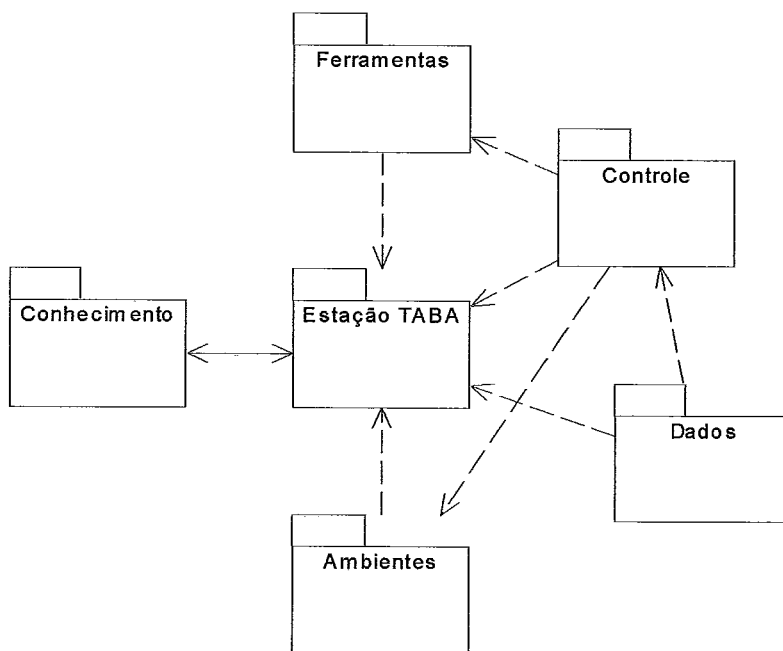


Figura 7.5 – Diagrama de Pacotes do Modelo da Estação TABA

Tabela 7.3 – Descrições do Conteúdo Atual dos Pacotes e das Alterações Realizadas

Pacotes	Conteúdo Atual:
Estação TABA	Classes responsáveis por fornecer serviços de infra-estrutura como acesso aos diversos elementos que podem ser utilizados na composição dos ambientes, aos próprios ambientes já criados, além de acesso ao conhecimento e aos dados necessários nos diversos ambientes. Alterações Realizadas: Criação de classes para gerência de dados e transferência de classes que se encontravam no pacote de Conhecimento.
Ambientes	Classes que representam os ambientes criados a partir da Estação TABA. Alterações Realizadas: Definição e alteração de classes e associações para contemplar os novos tipos de ambientes.
Conhecimento	Classes responsáveis pela representação do conhecimento adquirido e disponibilizado nos ambientes da Estação TABA. Alterações Realizadas: Definição e alteração de classes e associações para contemplar os conhecimentos estabelecidos pela ontologia de organização e para possibilitar a composição de uma subteoria do domínio em termos de outras subteorias; associação de um conceito com a classe que será utilizada para representar as instâncias do conceito; e eliminação de especializações que tornavam o modelo desnecessariamente mais complexo.
Dados	Classes que organizam os dados utilizados e registrados nos ambientes da Estação TABA. Alterações Realizadas: Definição de classes e associações para contemplar os dados provenientes da ontologia de organização e necessários à descrição de uma organização; e transferência de classes que se encontravam nos pacotes de Conhecimento e de Controle.
Ferramentas	Classes que representam as ferramentas internas e externas.
Controle	Classes responsáveis pelo controle da execução das atividades do processo de software e pelo controle da gerência da qualidade dos produtos nos ambientes instanciados. Alterações Realizadas: Reorganização das classes para considerar os conceitos provenientes da ontologia de organização que são necessários à alocação de recursos às atividades e ao controle dos responsáveis pela execução das mesmas.

Cabe ressaltar que o modelo de classes da Estação TABA está em constante atualização na medida em que novas ferramentas foram e estão sendo agregadas à Estação TABA [25,27,28,29,30,31,35,37].

7.8 Ferramentas Desenvolvidas

Duas ferramentas foram desenvolvidas, como parte deste trabalho, para apoiar funcionalidades previstas na figura 7.2: uma ferramenta para apoiar a configuração de ambientes para as organizações e uma ferramenta para apoiar a descrição dos processos organizacionais nos Ambientes Configurados. Estas ferramentas são apresentadas a seguir (seções 7.8.1 e 7.8.2) e uma descrição detalhada de suas funcionalidades pode ser encontrada no anexo VIII.

➤ Ferramenta para Configuração do Meta-ambiente TABA

Para apoiar a configuração de ambientes para as organizações, foi definida e implementada a ferramenta *Config*, a ser utilizada pelo configurador de ambientes. A ferramenta é orientada a processo, possuindo uma barra na lateral esquerda com a representação explícita do processo de configuração. A ferramenta permite, portanto,

a contextualização da configuração desejada, a elaboração de uma proposta de configuração com o posterior registro de um parecer sobre a mesma, a definição do processo padrão e dos processos especializados para a organização, a definição das Teorias de Domínio e das Descrições de Tarefas desejadas no Ambiente Configurado e, por fim, a criação do Ambiente Configurado. A figura 7.6 mostra a tela inicial do Config, que exibe a primeira atividade do processo a ser executada: *Realizar Entrevistas*, que é uma sub-atividade da atividade *Contextualizar Configuração*.

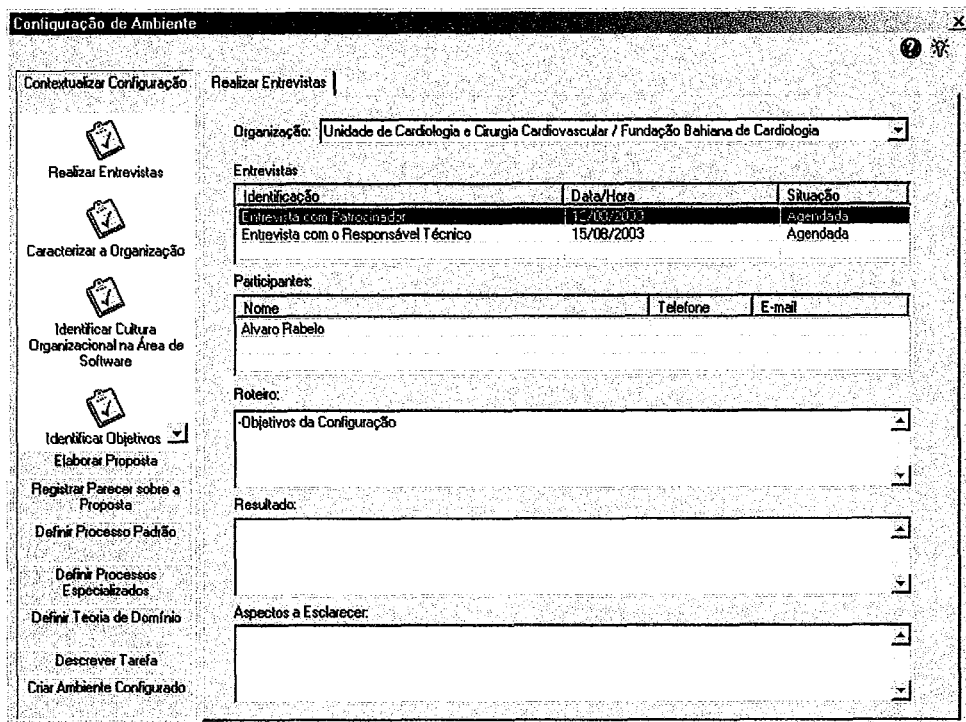


Figura 7.6 – Tela Inicial do Config: Realização de Entrevistas

A barra na lateral esquerda da figura 7.6 informa, em fundo amarelo, quais são as outras sub-atividades que compõem a atividade *Contextualizar Configuração* e, em fundo cinza, quais são as próximas atividades a serem executadas, orientando, assim, o configurador de ambientes. Assim como outras ferramentas desenvolvidas para apoiar a construção de ADSOrg Instanciados ou disponíveis nos ADSOrg Instanciados, a ferramenta *Config* possui dois ícones na lateral direita superior que têm a finalidade de, respectivamente, fornecer o conhecimento disponível no ambiente para auxiliar a execução da atividade (🔍) e executar o módulo da ferramenta *Acknowledge* [31] que permite o registro de conhecimento obtido na execução da atividade (📝).

No final deste capítulo, é apresentado um exemplo de uso da ferramenta *Config* na configuração de um ambiente para uma organização.

As classes adicionadas ao modelo da Estação TABA para implementação da ferramenta Config também são apresentadas no anexo VII, que contém o atual modelo de classes da Estação TABA.

➤ Ferramenta para Descrição dos Processos Organizacionais

Para apoiar a descrição dos processos organizacionais nos Ambientes Configurados e fornecer este conhecimento para orientar a execução das atividades dos desenvolvedores de *software* nos ADSOrg Instanciados, possibilitando, também, a sua atualização e evolução, foi definida e implementada a ferramenta ProcKnow. Além de possibilitar a descrição e visualização dos processos executados na organização, sejam estes processos de software ou não, a ferramenta tem como objetivo descrever e fornecer o contexto em que um determinado conhecimento é utilizado, de forma a facilitar o entendimento tanto da atividade quanto dos conhecimentos necessários a sua execução. Sendo assim, ProcKnow provê suporte ao uso da linguagem de modelagem de processos proposta no capítulo 5, permitindo:

- a representação gráfica dos processos organizacionais, com a definição de quais são as atividades que compõem um processo, como elas estão relacionadas quanto à ordem de execução e quais são os atores envolvidos. A cada atividade podem ser associados os seus insumos e produtos, como também quais são os recursos materiais e as competências requeridas para a sua execução;
- a descrição das atividades organizacionais, o que pode envolver a representação gráfica de como essas atividades estão organizadas em termos de sub-atividades, dando origem a um conjunto aninhado de diagramas que, juntos, representam um processo;
- a descrição do ator, oferecendo a possibilidade de associá-lo a uma posição ou cargo da estrutura organizacional, a uma unidade organizacional ou, ainda, a um grupo de posições, cargos ou pessoas da organização;
- a descrição das matérias-primas e artefatos que representam os insumos e os produtos de uma atividade, dos bens de produção que atuam como recursos e dos conhecimentos requeridos e produzidos. Um conhecimento explícito disponível no ambiente deve poder ser associado a sua representação em um diagrama;
- a visualização dos processos organizacionais e a navegação através dos seus diferentes níveis de abstração, fornecendo, sob solicitação, detalhes sobre os elementos representados nos diagramas e permitindo acesso a conhecimento formalizado.

A figura 7.7 mostra a tela principal da ferramenta Procknow, que se divide em cinco seções: 1) uma barra com *menus* que organizam as operações específicas da ferramenta e as operações comuns ao padrão Microsoft de *interface* com o usuário; 2) uma barra de ferramentas para acesso rápido a várias dessas operações; 3) uma barra com os elementos gráficos³ disponíveis para a modelagem dos processos; 4) uma janela com a árvore de diagramas que compõem o modelo sendo editado, que pode ser um novo modelo ou um modelo já existente; e 5) uma janela com o diagrama selecionado para edição. Vários modelos podem ser abertos simultaneamente, mas, ao iniciar-se a execução da ferramenta, as seções 4 e 5 referem-se a um novo modelo, que é provisoriamente intitulado *NovoModelo*. Um modelo de processos pode ser composto de vários diagramas e, como mencionado anteriormente, também pode conter conjuntos aninhados de diagramas, o que ocorre quando uma atividade ou processo é detalhado em um diagrama de nível mais baixo, ou seja, é decomposto. Na figura 7.7, cada uma dessas seções encontra-se identificada.

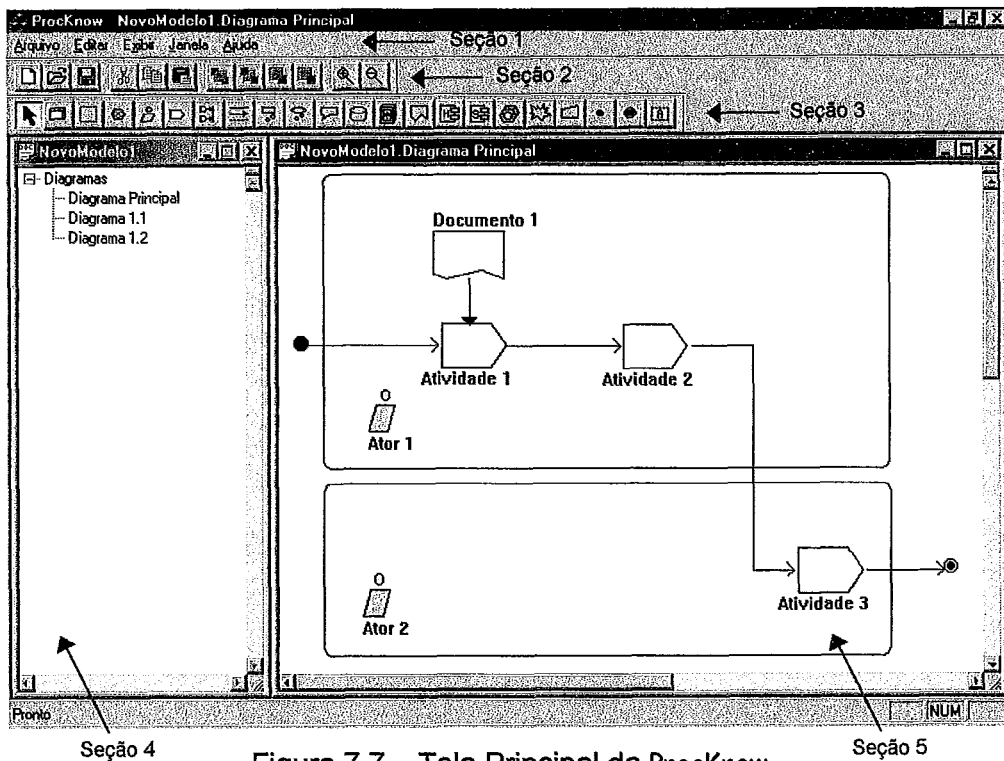


Figura 7.7 – Tela Principal da Procknow

Em termos de implementação, a ferramenta Procknow está baseada no trabalho realizado por FATEEV [210] para construção de um meta-gerador de ferramentas para apoio à elaboração de modelos gráficos referentes aos métodos de desenvolvimento. Desta forma, o modelo de classes definido contém um conjunto de classes pertencentes ao nível meta que têm como finalidade possibilitar a descrição dos

³ Elementos gráficos são objetos ou ligações gráficas.

elementos que definem a ferramenta, as quais são: meta-modelo, meta-diagrama, meta-elemento (meta-objeto ou meta-ligação), meta-atributo, regra de agrupamento, regra de ligação. Para armazenar e manipular os modelos de processos elaborados pelo usuário, foram definidas as classes: modelo, diagrama, elemento gráfico (objeto gráfico e ligação gráfica), objeto lógico, associação lógica e atributo. A intenção é que o código da ferramenta ProcKnow seja utilizado, em trabalho futuro, como base para definição do código a ser gerado por um meta-gerador de ferramentas baseado no mesmo modelo de classes e capaz de instanciar ferramentas específicas para apoio a diferentes modelos gráficos. As classes do nível meta dão origem às classes que representam os tipos dos elementos permitidos nas ferramentas específicas. A ferramenta ProcKnow já verifica os tipos dos elementos e as possibilidades de ligação e de agrupamento entre eles de forma dinâmica. A existência de um meta-gerador de ferramentas com base no modelo proposto dará suporte automático à evolução da linguagem de modelagem de processos, uma vez que alterações na linguagem acarretarão apenas a re-geração da ferramenta ProcKnow. O anexo IX apresenta o modelo de classes da ferramenta.

7.9 Estado Atual da Estação TABA

Como mencionado no capítulo 1, as idéias propostas neste trabalho deram origem a trabalhos de final de curso de graduação e de mestrado, quase todos já concluídos, que têm complementado o esforço de extensão da Estação TABA para implementar o conceito de ADSOrg.

Neste contexto, as seguintes ferramentas foram desenvolvidas:

- Sapiens, ferramenta que permite a descrição e visualização de estruturas organizacionais, englobando os profissionais alocados e as competências requeridas e possuídas ao longo dessas estruturas [25];
- AdaptPro, ferramenta para apoiar a instanciação de processos de software para projetos específicos a partir dos processos especializados da organização [30];
- Acknowledge, ferramenta para apoiar o processo de captura de conhecimento ao longo dos processos de software, englobando registro, filtragem e empacotamento de conhecimento [31];
- DocPlan, ferramenta para apoiar o planejamento da documentação a ser produzida em um projeto de software [37];
- TempPlan, TempManager, CustPlan e CustManager, ferramentas para apoiar o planejamento e o controle de tempo e custo em projetos de software, baseadas na

reutilização do conhecimento organizacional e nos modelos paramétricos COCOMO II e Análise de Pontos de Função [28];

- *RHPlan* e *RHManager*, ferramentas para apoiar o planejamento, monitoração e avaliação da alocação de profissionais aos projetos de software, que inclui a solicitação de contratação e capacitação, o acompanhamento das horas dedicadas a cada atividade, além da atualização, ao final do projeto, das competências por eles possuídas [29];
- *RiscManager*, ferramenta para apoiar o planejamento e a monitoração de riscos em projetos de software que baseia-se na reutilização do conhecimento organizacional sobre riscos [27];
- *GConf*, ferramenta para apoio à gerência de configuração dos artefatos produzidos em um projeto de software [35].

Além disso, outras ferramentas para planejamento de projetos de software também foram desenvolvidas no contexto do projeto TABA, as quais são:

- *OrgPlan*, ferramenta para elaboração do plano de organização para um projeto;
- *QualityPlan*, ferramenta para apoiar o planejamento do controle da qualidade que deve ser efetuado no projeto;
- *ControlManager*, ferramenta para apoiar o planejamento do acompanhamento e controle do projeto;
- *MedPlan*, ferramenta para apoiar o planejamento das medições no contexto da organização e do projeto;
- *Metrics*, ferramenta para apoiar a coleta de métricas baseada no plano de medição;
- *ActionPlanManager*, ferramenta para apoiar a elaboração de planos de ação ao longo do projeto;
- *Planilha de Atividades*, ferramenta para registro das atividades do desenvolvedor no projeto;
- *GeraDoc*, ferramenta para gerar documentos a partir da agregação de outros documentos;
- *ProjectStatus*, ferramenta que informa a situação dos projetos sendo conduzidos pela organização.

O conjunto atual de ferramentas da Estação TABA é composto pelas ferramentas mencionadas acima, pelas duas ferramentas apresentadas neste capítulo e por ferramentas já existentes na versão C++ da Estação TABA. A tabela 7.4 estabelece a relação entre as funcionalidades presentes na figura 7.2 (*Principais*

Componentes do Meta-ambiente, Ambientes Configurados e ADSOrg Instanciados) e o conjunto atual de ferramentas da Estação TABA.

Com relação à tabela 7.4, deve-se destacar que não são listadas as funcionalidades da figura 7.2 que são implementadas por serviços, ou que compõem uma outra funcionalidade (são sub-funcionalidades) e não estão disponíveis de forma independente. Além disso, as funcionalidades *Instancia ADSOD* e *Instancia ADS Convencional* estão contempladas na Estação TABA através do apoio existente a suas sub-funcionalidades.

Tabela 7.4 – Relação entre Funcionalidades e Ferramentas

Ambiente	Funcionalidade	Ferramentas
Meta-ambiente	Configura Ambiente	Config
	Define Teoria do Domínio	EdiTeD
	Descreve Tarefa	EdiTAr
	Planeja Processo	Edit-Pro e Assist-Pro
Ambiente Configurado	Descreve Organização	Sapiens
		ProcKnow
	Gerencia Conhecimento Organizacional	EdiTeD
		Genesis
		Navegue
		Acknowledge
MedPlan		
Instancia ADSOrg	AdaptPro	
ADSOrg Instanciado	Orienta a Execução da Atividade	Sapiens
		ProcKnow
		Navegue
		Genesis e RegCon
		Acknowledge
	Executa Atividade	Q-Fuzzy
		Planilha de Atividades
	Orienta a Execução da Atividade e Executa Atividade	TempPlan e TempManager
		CustPlan e CustManager
		RHPlan e RHManager
		RiscManager
		OrgPlan
		DocPlan
		QualityPlan
		ControlManager
		MedPlan e Metrics
		ActionPlanManager
	ProjectStatus	
	Gerencia Artefatos	Gconf
		DocPlan
GeraDoc		
	Gconf	

Comparando as ferramentas mencionadas na descrição do Projeto TABA (capítulo 4) com as ferramentas que constam da tabela 7.4, percebe-se que a ferramenta Def-Pro foi substituída. O apoio fornecido pela ferramenta Config à definição de processos padrão e à especialização dos mesmos é uma evolução das idéias presentes na ferramenta Def-Pro.

A tabela 7.5 indica, para cada requisito listado na seção 7.4, se o mesmo encontra-se contemplado, parcialmente contemplado ou não contemplado na versão atual da Estação TABA.

Tabela 7.5 – Situação dos Requisitos definidos para os Ambientes da Estação TABA

Grupo Requisitos	Requisito	Situação
Requisitos gerais	R1	Contemplado
	R2	Contemplado
	R3	Contemplado parcialmente
	R4	Contemplado parcialmente
	R5	Contemplado parcialmente
Requisitos comuns ao Meta-Ambiente e aos Ambientes Configurados	R6	Contemplado
	R7	Não contemplado
	R8	Contemplado
Requisitos específicos do Meta-Ambiente	R9	Contemplado
	R10	Contemplado
	R11	Contemplado
	R12	Não contemplado
	R13	Contemplado
	R14	Contemplado
	R15	Não contemplado
	R16	Não contemplado
Requisitos específicos dos Ambientes Configurados	R17	Contemplado
	R18	Contemplado
	R19	Contemplado
	R20	Contemplado
	R21	Contemplado
	R22	Contemplado
	R23	Contemplado
Requisitos adicionais dos Ambientes Configurados para Organizações de Software	R24	Contemplado
	R25	Contemplado
	R26	Contemplado
	R27	Não contemplado
Requisitos comuns a todos ADS Instanciados	R28	Contemplado parcialmente
	R29	Contemplado
	R30	Contemplado parcialmente
	R31	Contemplado
Requisitos comuns a ADSOD e ADSOrg Instanciados	R32	Contemplado
	R33	Contemplado
Requisitos específicos dos ADSOrg Instanciados	R34	Contemplado
	R35	Contemplado
	R36	Contemplado
	R37	Contemplado
	R38	Contemplado
	R39	Contemplado
Requisitos adicionais dos ADSOrg Instanciados em Organizações de Software	R40	Contemplado
	R41	Contemplado
	R42	Contemplado

7.10 Cordis-FBC: um Ambiente Configurado TABA

Cordis-FBC é o ambiente que foi configurado para a Fundação Bahiana de Cardiologia (FBC), a partir do Meta-ambiente TABA, para avaliar a aplicabilidade e viabilidade da solução proposta. A FBC é uma organização que presta serviços de assistência ambulatorial e hospitalar na área de cardiologia, possuindo um Núcleo de

Pesquisa e Desenvolvimento (NPqD) que desenvolve diferentes tipos de *software* médico, tais como sistemas especialistas e *software* educacional para medicina, além de sistemas de informação hospitalar, de prontuário médico eletrônico e de telemedicina.

A motivação para a configuração do *Cordis-FBC* foi a freqüente perda de conhecimento adquirido pelos desenvolvedores do NPqD sobre o domínio e sobre a organização, devido à alta rotatividade de pessoal no setor, uma característica comum em ambientes acadêmicos de pesquisa e desenvolvimento. Neste contexto, profissionais da área de saúde freqüentemente tinham que explicar novamente um conceito do domínio ou um processo da organização para um desenvolvedor recém-chegado à FBC e tempo demais era gasto na identificação de quem na organização detinha o conhecimento necessário à execução de uma determinada atividade. Apesar do conhecimento sobre o domínio de cardiologia já estar embutido no ambiente *Cordis*, construído anteriormente para a FBC por OLIVEIRA [20], este ambiente não contemplava o conhecimento sobre a organização e sobre as experiências acumuladas pela mesma ao longo dos seus projetos de desenvolvimento de *software* e não previa a gerência do conhecimento embutido no ambiente.

Sendo assim, o processo de configuração proposto no anexo V foi executado com o apoio da ferramenta *Config*, de forma a configurar o ambiente *Cordis-FBC*. As figuras 7.8 a 7.21 exibem telas que ilustram a execução do processo de configuração.

Devido à experiência de 10 anos de trabalho conjunto da COPPE/UFRJ com a FBC, grande parte das informações a serem coletadas e registradas na atividade *Contextualizar Configuração* já eram conhecidas. Entrevistas foram realizadas para descobrir o tamanho atual da organização e os objetivos almejados pela organização com a configuração do ambiente *Cordis-FBC*. As figuras 7.8 a 7.12 exibem as informações referentes a contextualização da configuração.

Como não havia, na FBC, perspectiva de desenvolvimento de novos sistemas baseados em conhecimento ou de manutenção do sistema já existente, a proposta de configuração estabeleceu: 1) a inclusão, no Ambiente Configurado, da Teoria do Domínio de Cardiologia já definida por OLIVEIRA [20]; 2) a revisão do processo padrão de desenvolvimento da organização e a definição de um processo padrão de manutenção de acordo com a norma ISO/IEC 12207 [140] ; 3) a definição de um processo especializado para o paradigma estruturado; e 4) a utilização de ferramentas da Estação TABA e de ferramentas já disponíveis na organização. As figuras 7.13 e 7.14 exibem algumas telas referentes à elaboração da proposta para configuração do *Cordis-FBC*.

Configuração de Ambiente

Contextualizar Configuração

Caracterizar a Organização

Realizar Entrevistas

Caracterizar a Organização

Identificar Cultura Organizacional na Área de Software

Identificar Objetivos

Elaborar Proposta

Registrar Parecer sobre a Proposta

Definir Processo Padrão

Definir Processos Especializados

Definir Teoria de Domínio

Descrever Tarefa

Criar Ambiente Configurado

Caracterizar a Organização

Organizações:

Nome	Descrição
COPPE	Instituto Alberto Luiz Coimbra de Pós-graduação e Pes.
RioSoft	Micro e pequenas empresas de Desenvolvimento de s.
Unidade de Cardiologia e Cirurgia Cardiovascular / Flac.	Unidade do Hospital Invasivo da UFPA - AUCITV/3

Indústria:

Tamanho da Organização:

Domínios de Conhecimento do Negócio:

Nome
Cardiologia

Domínios de Conhecimento de Apoio ao Negócio:

Nome
Administração Hospitalar
Desenvolvimento de Software

Figura 7.8 – Tela de Caracterização da FBC

Configuração de Ambiente

Contextualizar Configuração

Identificar Cultura Organizacional na Área de Software

Realizar Entrevistas

Caracterizar a Organização

Identificar Cultura Organizacional na Área de Software

Identificar Objetivos

Elaborar Proposta

Registrar Parecer sobre a Proposta

Definir Processo Padrão

Definir Processos Especializados

Definir Teoria de Domínio

Descrever Tarefa

Criar Ambiente Configurado

Identificar Cultura Organizacional na Área de Software

Informações Gerais | Processos | Tecnologias

Número de Profissionais na Área de Software:

Certificações/Avaliações:

Nome	Tipo	Descrição
------	------	-----------

Tipos de Software quanto à Finalidade:

Nome
Software para uso próprio

Tipos de Software quanto à Tecnologia:

Nome
Software para Web

Tamanho dos Projetos:

Pequeno (<14 homem-mês)

Médio (<36,5 homem-mês)

Grande (>= 36,5 homem-mês)

Complexidade dos Projetos:

Baixa

Média

Alta

Software Crítico

Principais Problemas em Projetos de Software:

Figura 7.9 – Tela de Identificação da Cultura Organizacional da FBC na Área de Software: Seção de Informações Gerais

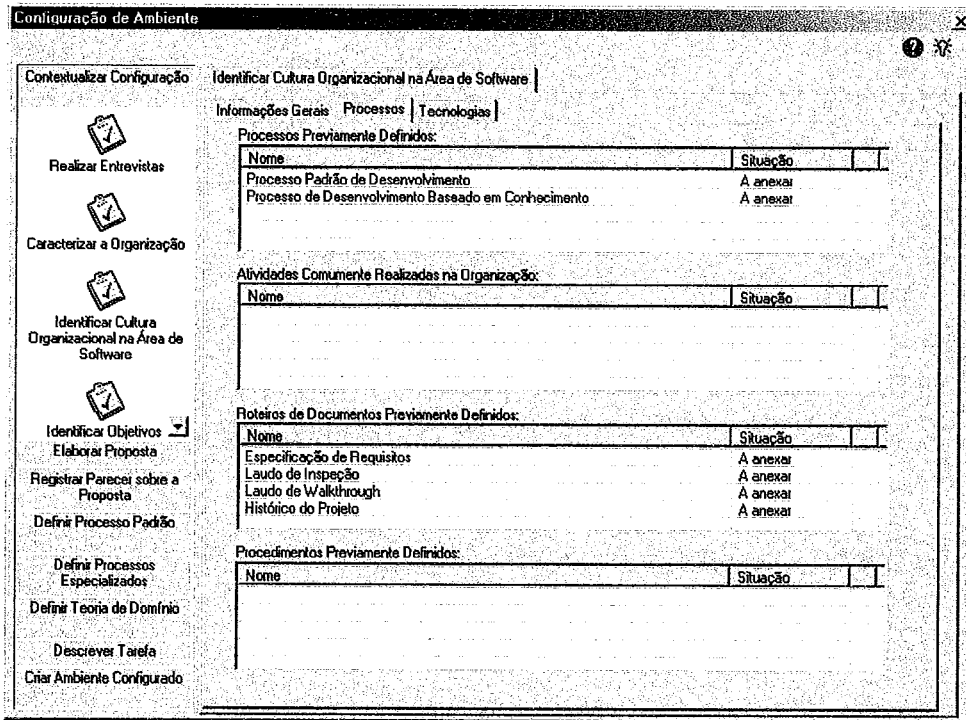


Figura 7.10 – Tela de Identificação da Cultura Organizacional da FBC na Área de Software: Seção de Processos

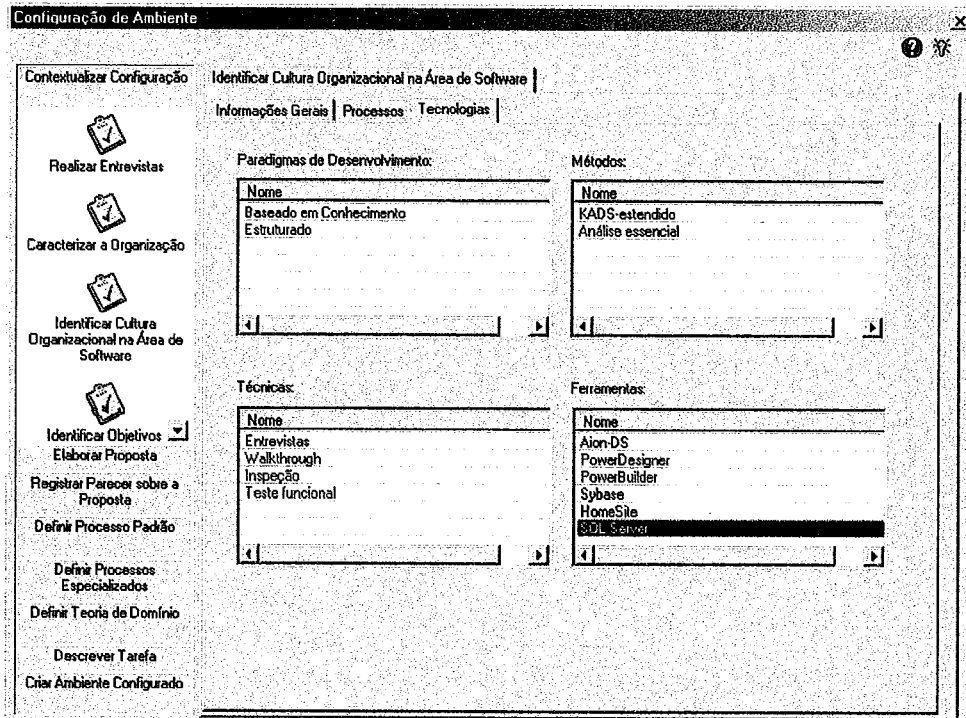


Figura 7.11 – Tela de Identificação da Cultura Organizacional da FBC na Área de Software: Seção de Tecnologias

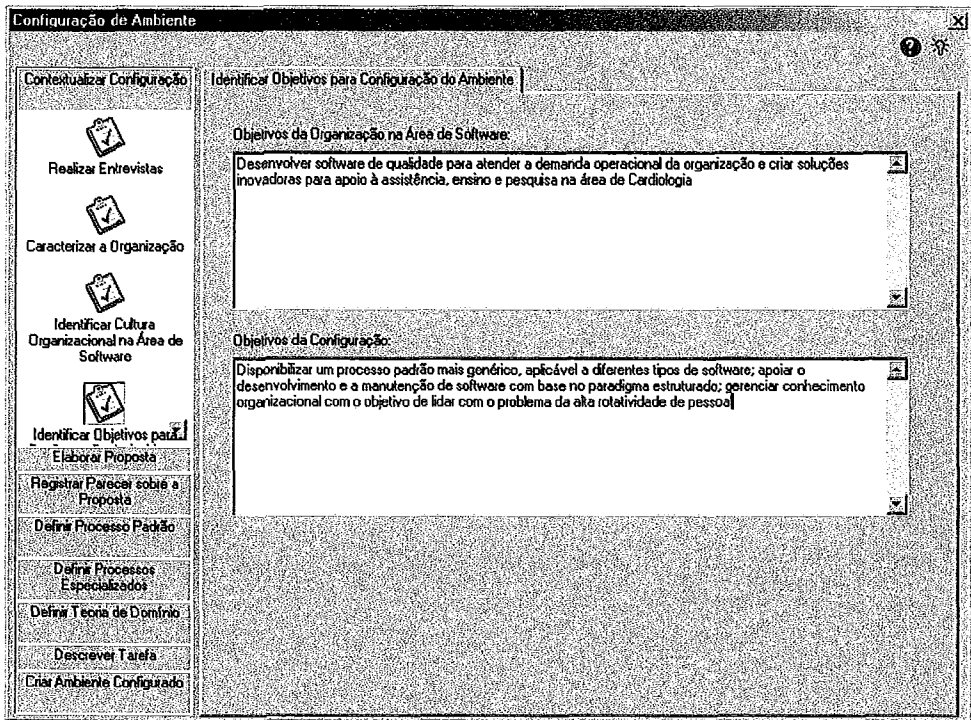


Figura 7.12 – Tela de Identificação dos Objetivos para Configuração do Cordis-FBC

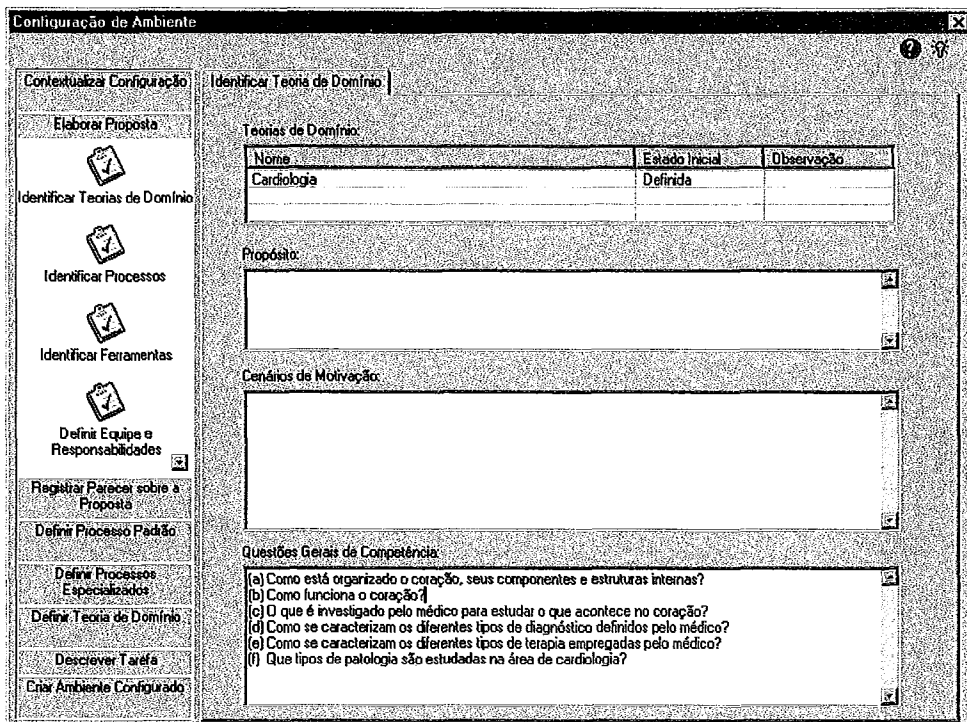


Figura 7.13 – Tela para Identificação das Teorias do Domínio a serem Fornecidas

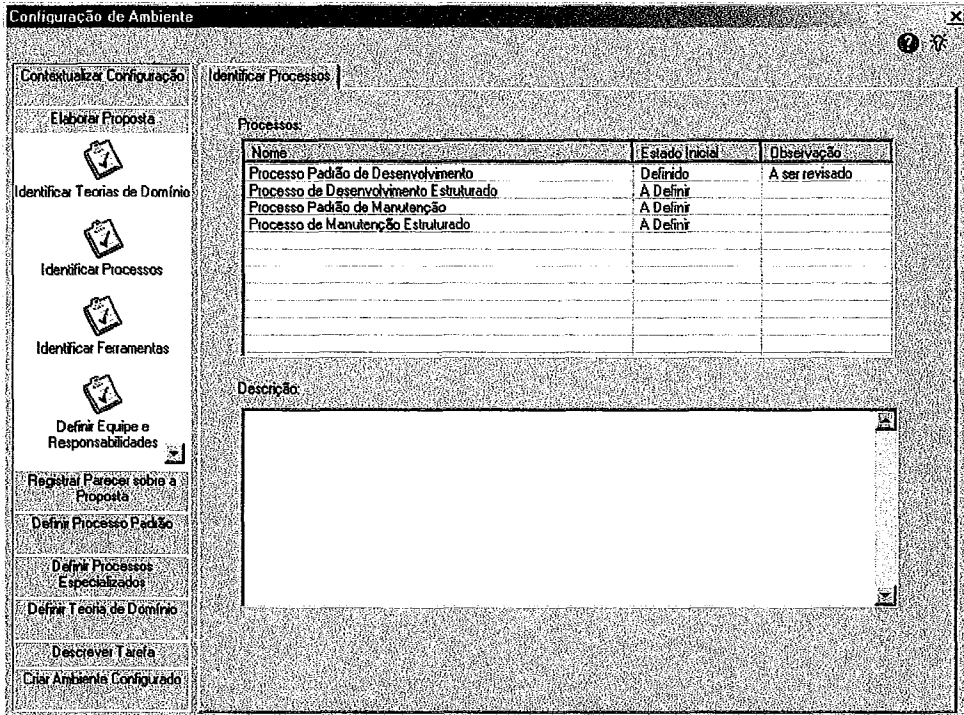


Figura 7.14 – Tela para Identificação dos Processos a serem Fornecidos

Uma vez aprovada a proposta de configuração, deu-se início à definição do processo padrão da organização, envolvendo a definição dos processos de ciclo de vida de desenvolvimento e de manutenção bem como a definição dos respectivos roteiros de documentos. As figuras 7.15 a 7.18 apresentam algumas telas referentes à definição do processo padrão.

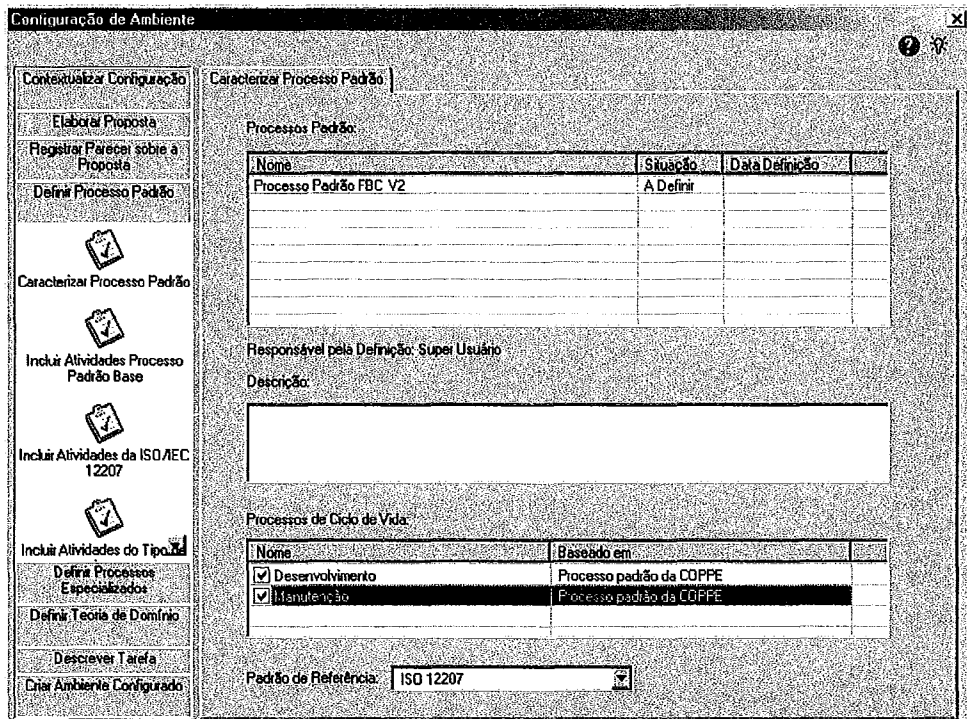


Figura 7.15 – Tela de Caracterização do Processo Padrão

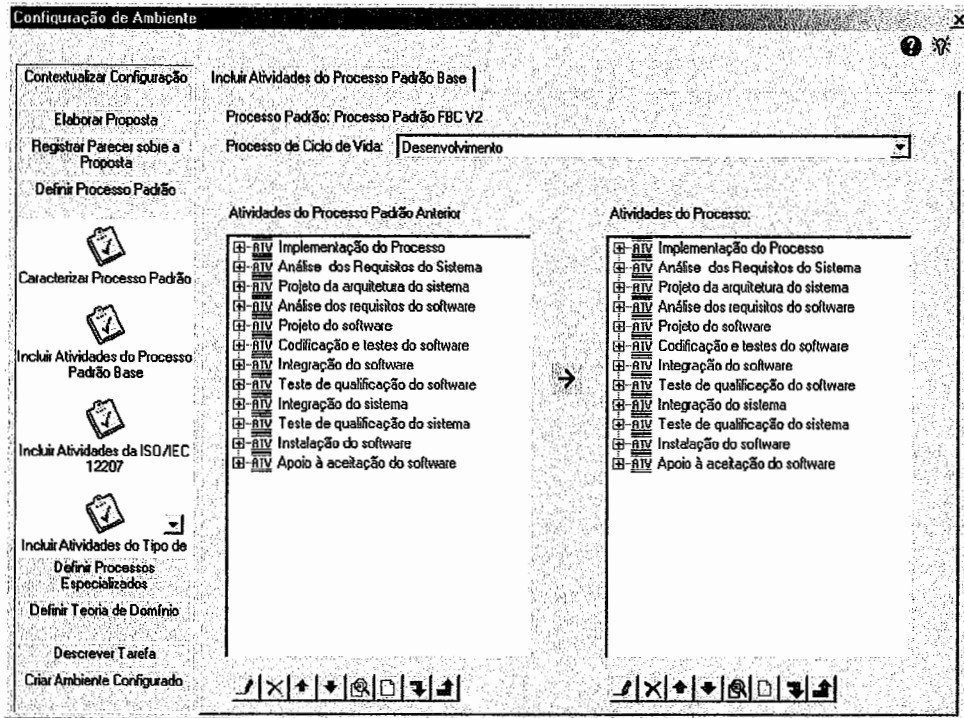


Figura 7.16 – Tela de Inclusão de Atividades do Processo Base

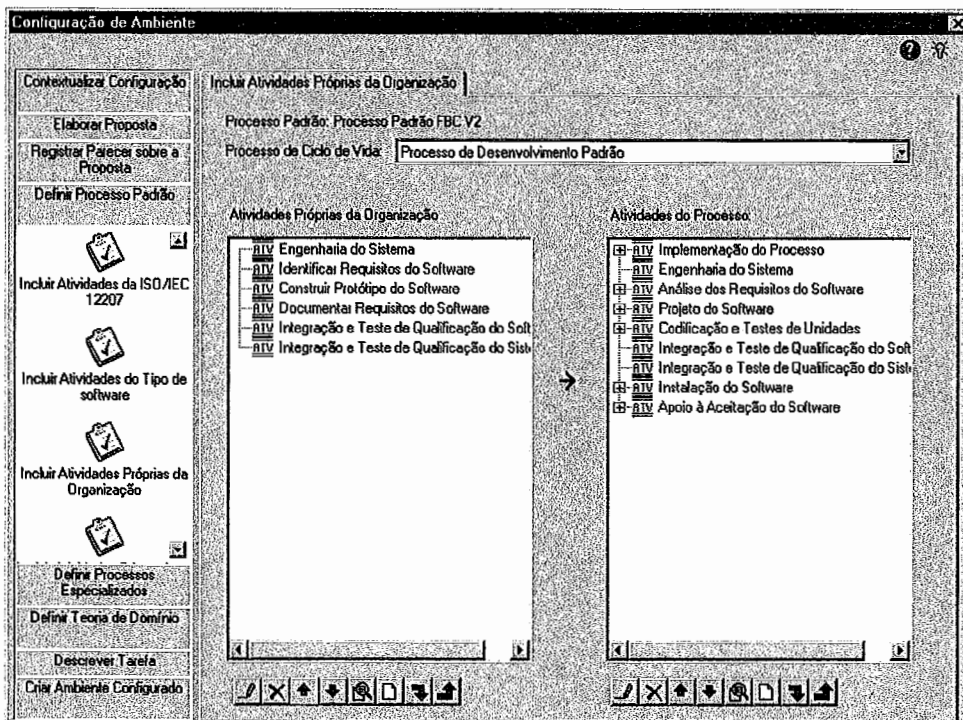


Figura 7.17 – Tela de Inclusão de Atividades Próprias da Organização

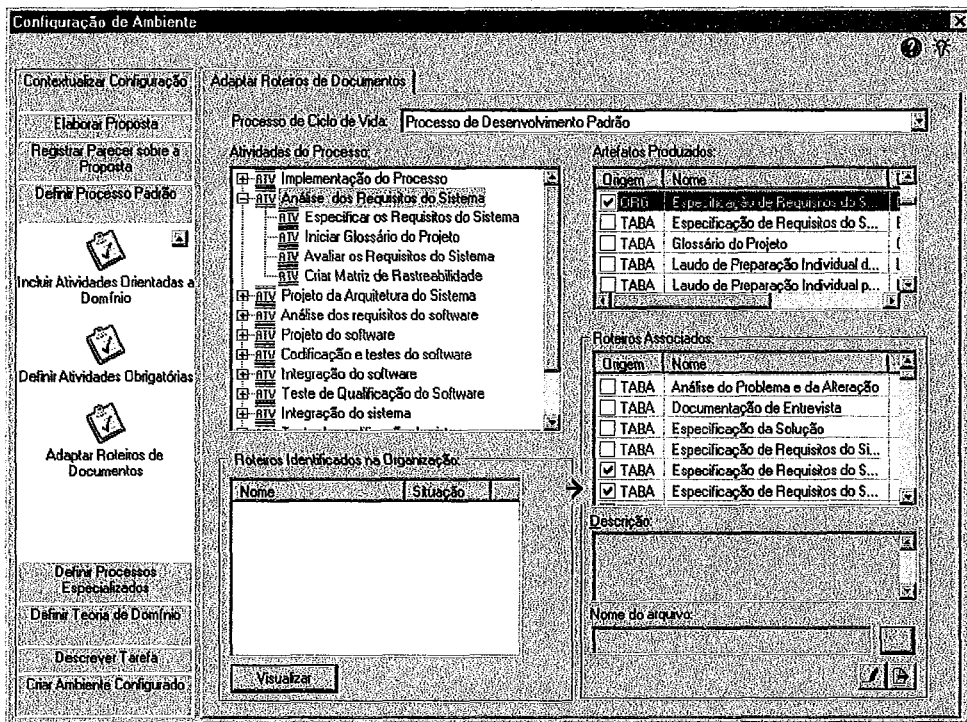


Figura 7.18 – Tela para Adaptação Roteiros de Documentos para a Organização

Com o processo padrão definido, partiu-se para a definição dos processos de ciclo de vida especializados de acordo com o paradigma estruturado e, conseqüentemente, para a especialização de alguns roteiros de documentos. A figura 7.19 mostra a possibilidade de inclusão, no processo especializado, de atividades próprias do desenvolvimento de software para Web. Para a FBC, no entanto, a inclusão de tais tipos de atividades foi postergada para o momento da instanciação de processos para projetos específicos.

Já as atividades de *Definir Teoria do Domínio* e *Descrever Tarefa* não precisaram ser executadas, pois, como mencionado anteriormente, a Teoria do Domínio de Cardiologia já estava definida e disponível na Estação TABA desde a construção do ambiente Cordis [20]. Também não foi proposto o desenvolvimento de novas ferramentas para o Cordis-FBC, de forma que a atividade *Incluir Novas Ferramentas nas Descrições dos Processos* também não precisou ser executada. Desta forma, o ambiente Cordis-FBC pôde, então, ser criado e entregue à FBC. As figuras 7.20 e 7.21 ilustram as sub-atividades de *Criar Ambiente Configurado* nas quais o ambiente foi definido e gerado. Para a definição de um Ambiente Configurado, são indicados o processo padrão e, implicitamente, seus respectivos processos especializados, bem como as teorias de domínio a serem incorporadas no ambiente.

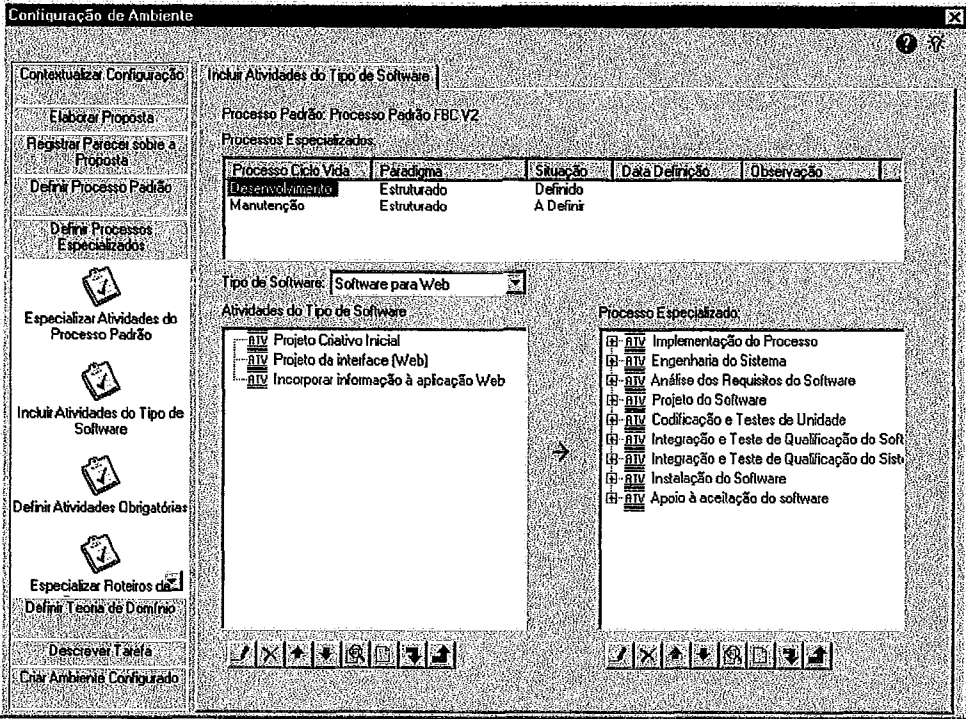


Figura 7.19 – Tela de Inclusão de Atividades do Tipo de Software

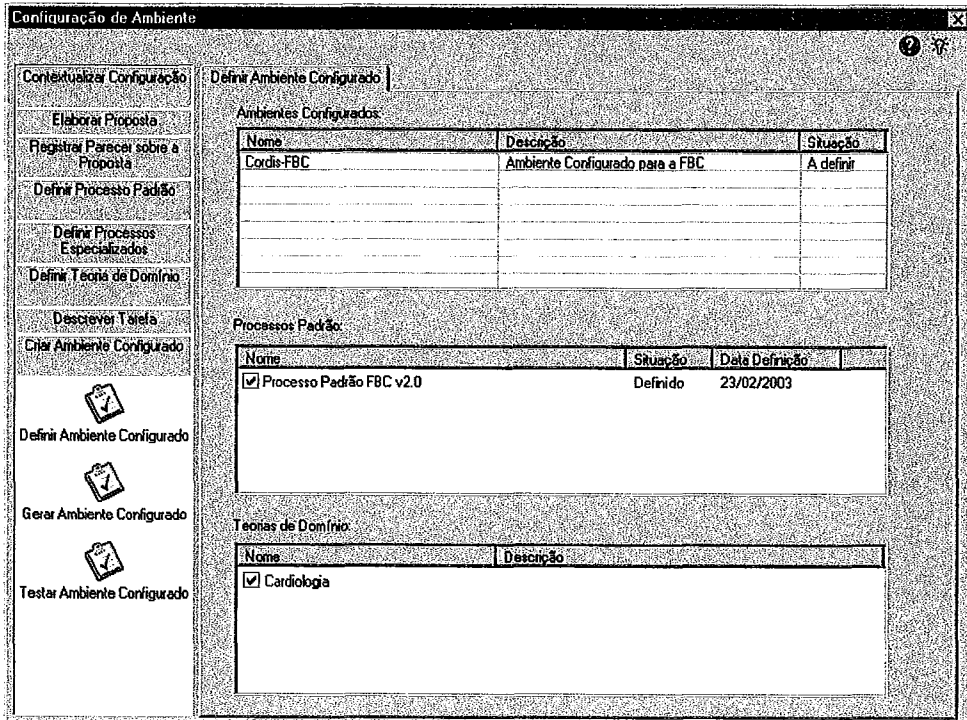


Figura 7.20 – Tela para Definição do Ambiente

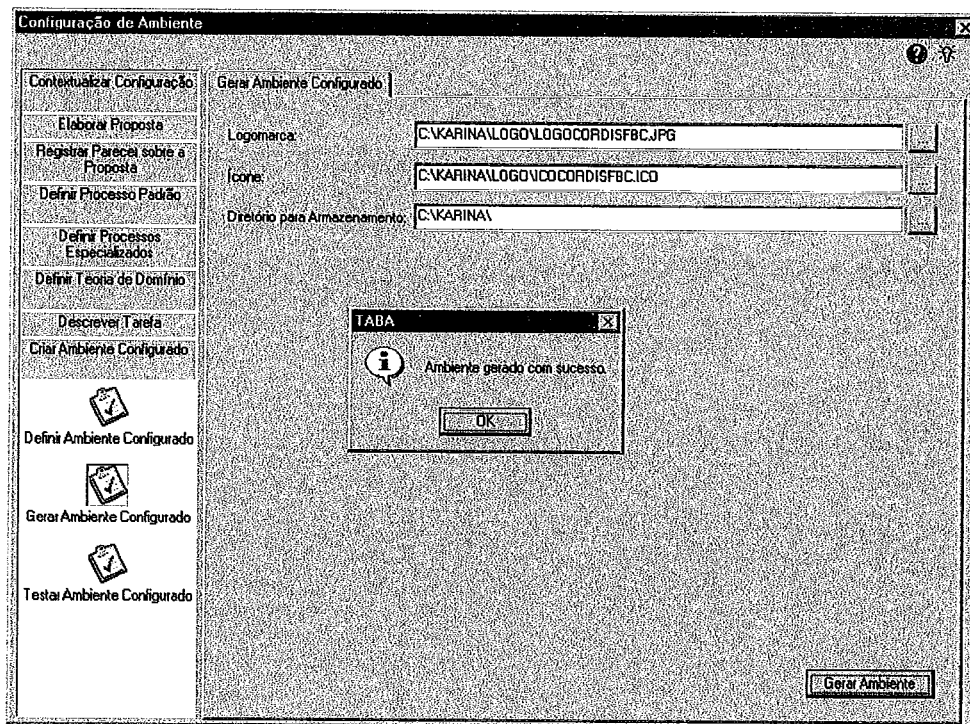


Figura 7.21 – Tela para Geração do Ambiente

A figura 7.22 apresenta a tela principal do ambiente configurado Cordis-FBC ao ser executado após a sua instalação na organização. A logomarca foi fornecida na atividade de geração do ambiente (figura 7.21).

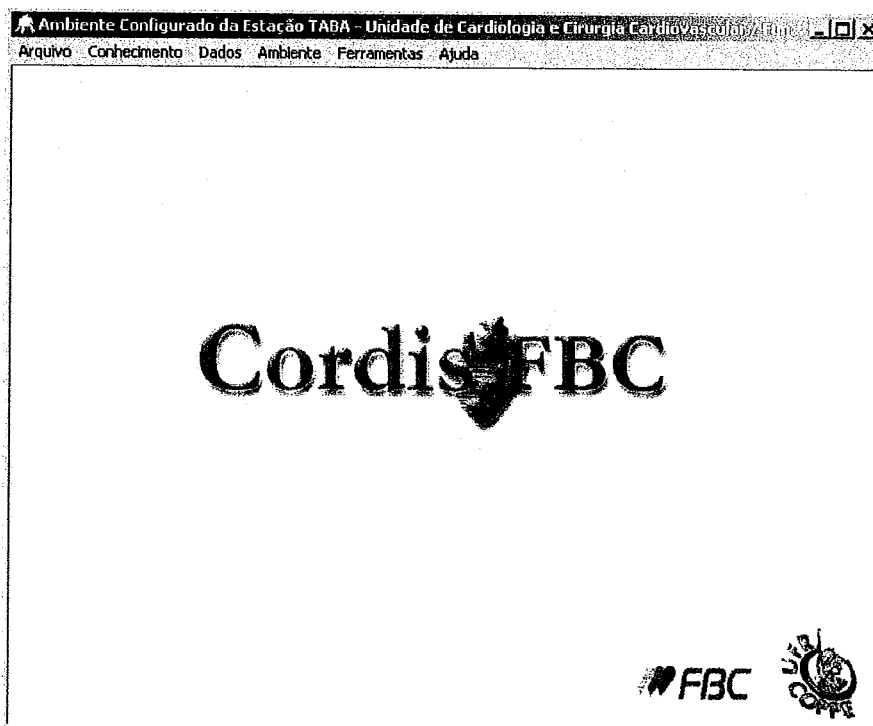


Figura 7.22 – Tela Principal do Cordis-FBC

O passo seguinte foi a preparação do Cordis-FBC para uso na organização através da descrição da estrutura e dos processos organizacionais, o que incluiu descrição da alocação de pessoal e das competências requeridas e possuídas ao longo da estrutura e dos processos organizacionais. O trabalho de levantamento desse conjunto de conhecimentos já havia sido realizado com o intuito de ser utilizado para validar a ontologia de organização e a linguagem de modelagem de processos, tendo envolvido, através de entrevistas, pessoas de várias áreas da organização. A preparação de um Ambiente Configurado segundo as atividades estabelecidas no anexo VI possibilita que conhecimento útil seja fornecido aos desenvolvedores desde o momento em que os Ambientes Configurado e Instanciados são disponibilizados.

As unidades organizacionais que compõem a FBC são: Presidência, Gerência Financeira, Setor de Contas a Receber, Setor de Contas a Pagar, Setor de Compras, Gerência de Engenharia e Segurança, Núcleo de Pesquisa e Desenvolvimento de Software Médico, Gerência Administrativa, Setor de Marcação, Setor de Recepção, Setor de Faturamento, Setor de Produtividade Médica, Gerência Médica, Gerência de Enfermagem, Setor de Cardiologia Clínica, Setor de Arritmia, Setor de Ergometria, Setor de Ergoespirometria, Setor de EcoDopplercardiografia, Setor de EcoDoppler Vascular, Setor de Lípidos, Setor de Medicina Nuclear, Setor de Hemodinâmica, Setor de Tomografia Computadorizada, Unidade de Internamento, Unidade Coronariana, Setor de Pronto Atendimento, Setor de Cirurgia Cardíaca, Laboratório de Pesquisas Cirúrgicas, Centro Cirúrgico, Farmácia, Setor de Anestesia, Setor de Fisioterapia, Setor de Raio X, Laboratório e uma série de ambulatórios. Cada uma dessas unidades foi descrita em termos de nome, descrição, objetivos, posições existentes, pessoas alocadas e relação de subordinação e de assessoria com outras unidades organizacionais.

A figura 7.23 mostra como a ferramenta Sapiens [25] é chamada e utilizada, por exemplo, para a descrição de uma unidade organizacional da FBC. Esta ferramenta foi implementada com base na ontologia de organização, ou seja, suas classes e associações foram derivadas dos conceitos e relações definidos pela ontologia, com o acréscimo de classes e associações resultantes de decisões de projeto.

Na figura 7.24, a ferramenta Sapiens está sendo utilizada para navegar ao longo da estrutura organizacional previamente descrita. Os nós da árvore hiperbólica representam a organização, suas unidades organizacionais, as posições existentes em cada unidade organizacional, as pessoas alocadas a essas posições e as competências associadas. Ao selecionar um nó na árvore, esse nó é posto em

destaque no centro da figura e detalhes sobre o mesmo são fornecidos no lado direito da tela. Na figura 7.24, são fornecidos os detalhes sobre a própria organização.

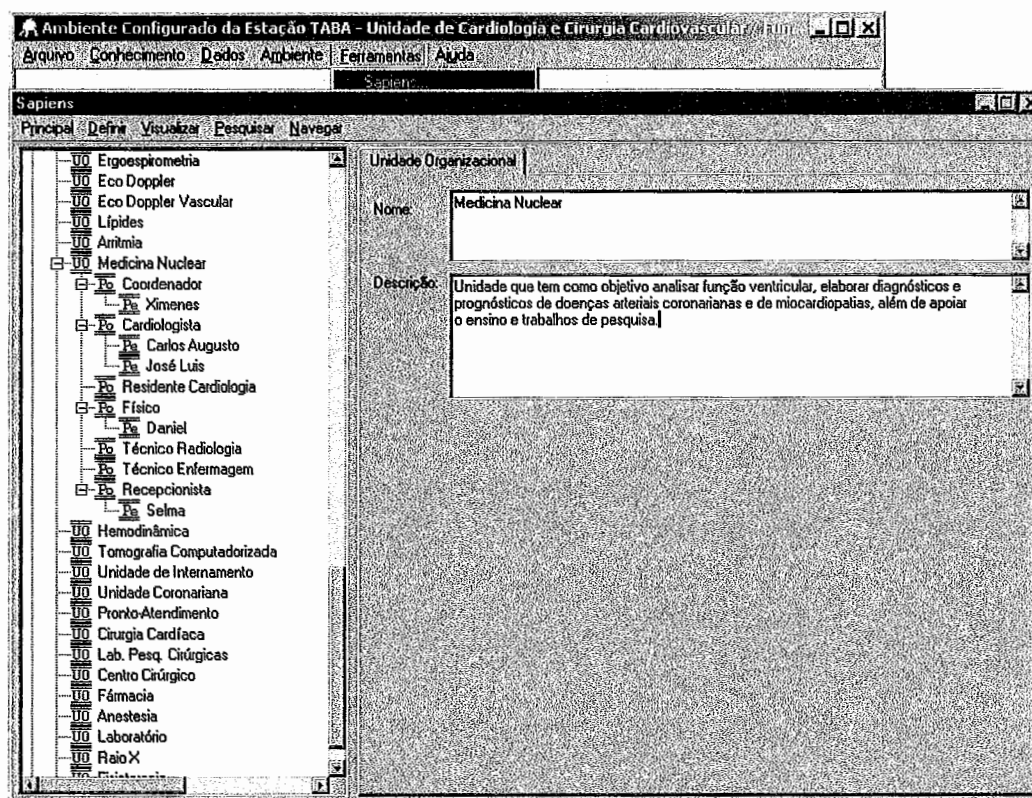


Figura 7.23 – Chamada da Ferramenta Sapiens e Tela com a Descrição de uma Unidade Organizacional da FBC

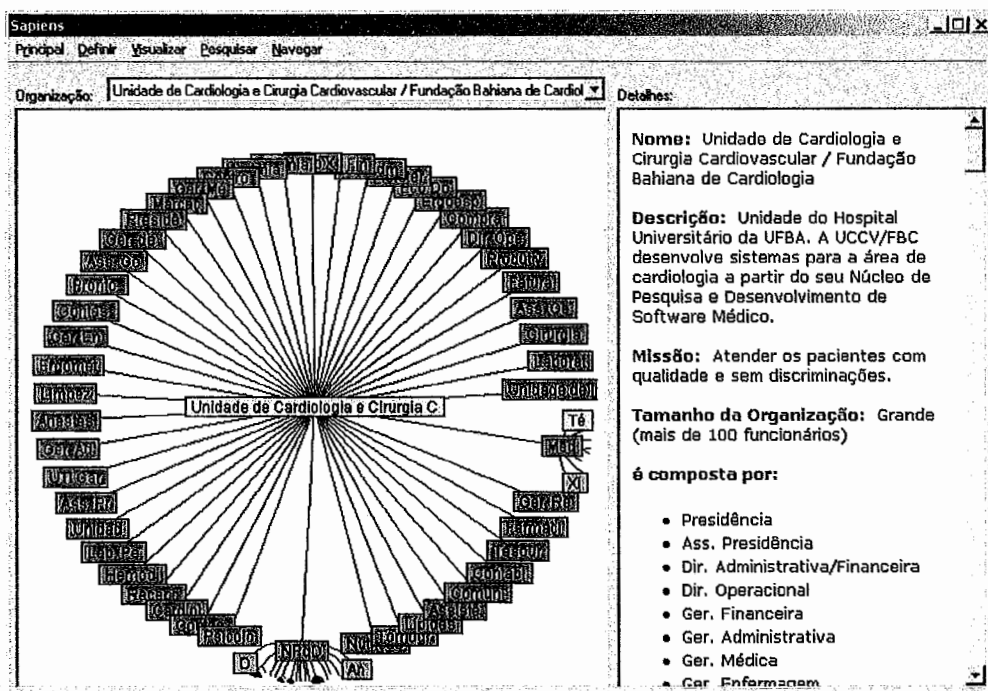


Figura 7.24 – Tela com a Representação Gráfica da Estrutura Organizacional da FBC

A figura 7.25 mostra como a ferramenta ProcKnow é chamada e utilizada para a descrição dos processos organizacionais da FBC, mais especificamente, para a descrição de um documento recém inserido em um diagrama. Nas figuras 7.26 e 7.27, a ferramenta ProcKnow está sendo utilizada para visualizar diagramas do modelo de processos da organização. Na figura 7.26, é apresentado o diagrama que contém os processos executados na organização, que são: Internamento Eletivo, Internamento de Emergência, Recebimento de Guia de Internamento, Marcação de Atendimento, Atendimento Ambulatorial, Entrega de Laudo, Recebimento de Material/Medicamento, Distribuição de Material/Medicamento e Compra de Material/Medicamento. O processo de *Distribuição de Material/Medicamento* pode ser ativado tanto pelo evento *Pedido de Material/Medicamento* quando em função do recebimento de um material ou medicamento cujo pedido estava pendente. Da mesma forma, o processo de *Compra de Material/Medicamento* pode ser ativado tanto pelo evento *Estoque insuficiente* quando em função da necessidade de distribuição de um material ou medicamento acima da quantidade disponível no estoque. A figura 7.27 apresenta o diagrama que detalha o processo de atendimento ambulatorial do tipo consulta médica.

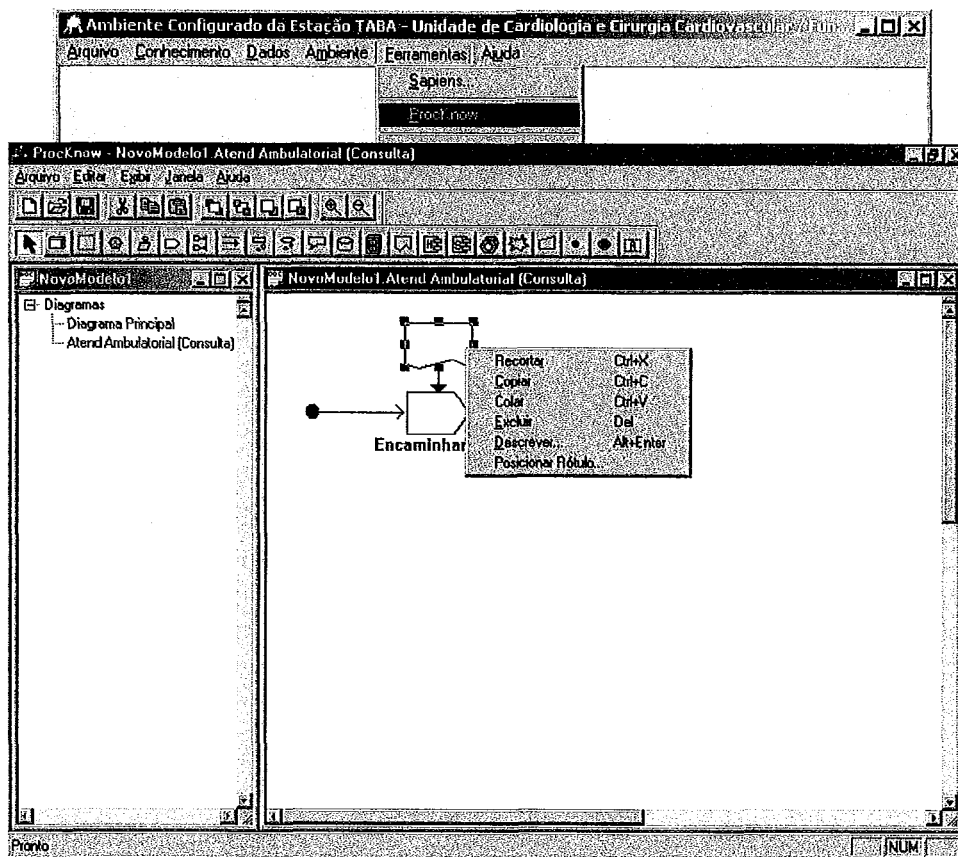


Figura 7.25 – Chamada da Ferramenta ProcKnow e Tela com a Edição de um Diagrama

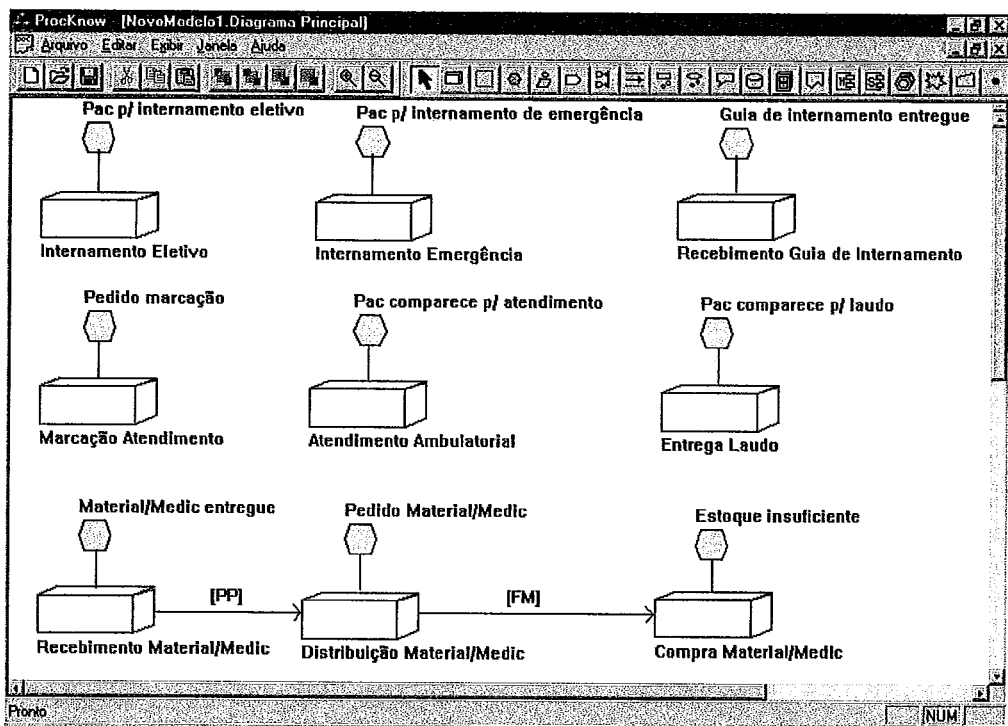


Figura 7.26 – Tela com Processos Executados na FBC

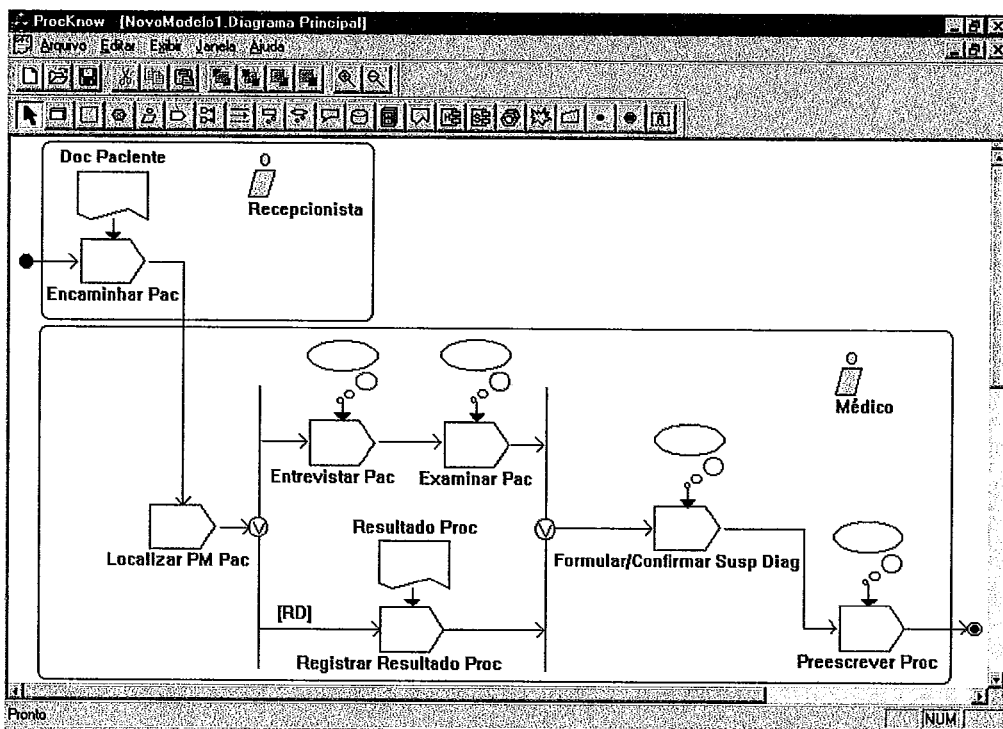


Figura 7.27 – Tela com o Detalhamento do Processo de Atendimento Ambulatorial (Consulta)

Outra atividade da preparação de um Ambiente Configurado para uso em uma organização é a captura do conhecimento sobre desenvolvimento e manutenção de software que já estava disponível para a organização, antes da configuração, em algum formato, local e meio de armazenamento. Na FBC, não havia conhecimentos

dessa natureza disponíveis para a organização, a não ser a descrição do processo de desenvolvimento e os roteiros de documentos, que já tinham sido capturados durante o processo de configuração do ambiente. A ferramenta utilizada seria a Acknowledge [31]. É importante ressaltar que não só as ferramentas Sapiens [25], ProcKnow, Acknowledge [31], mas também as ferramentas EdiTed [20], Genesis [148] e Navegue [148], podem ser utilizadas a qualquer momento para a gerência do conhecimento armazenado no Cordis-FBC.

A figura 7.28 ilustra a utilização da ferramenta Navegue para examinar o conteúdo da Teoria do Domínio de Cardiologia, mostrando, no lado esquerdo da tela, as subteorias (por exemplo, patologia e anatomia do coração), conceitos (por exemplo, função do coração e substrato anatômico) e conceitos especializados (por exemplo, parede e câmara cardíaca). Já no lado direito da tela, são exibidos os detalhes sobre a subteoria, conceito ou conceito especializado selecionado.

A Teoria do Domínio de Engenharia de Software é formada, por enquanto, apenas pela subteoria de Processo de Software e a instanciação dessa subteoria é feita através de um serviço do ambiente, uma vez que deriva classes responsáveis pelo controle nos ambientes da Estação TABA.

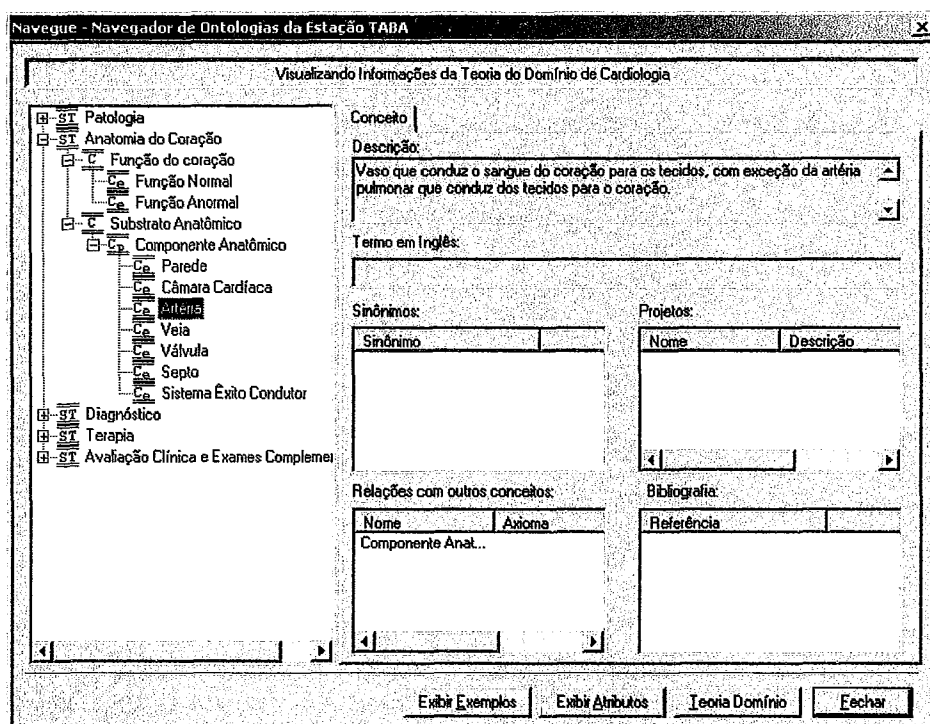


Figura 7.28 – Uso da Ferramenta Navegue para Visualização da Teoria do Domínio

Ao iniciar-se um projeto de software, o seu gerente utiliza a opção *Instanciar ADSOrg* do menu *Ambientes* do Cordis-FBC, chamando a ferramenta *AdaptPro* [30]. A ferramenta apóia as atividades do processo de instanciação de um ADSOrg para um

projeto específico, que são: caracterizar o projeto, planejar o processo e instanciar o ADSOrg. A figura 7.29 ilustra a sub-atividade *Definir Projeto*, que é parte da atividade *Caracterizar Projeto*. A figura 7.30 mostra a tela em que um modelo de ciclo de vida é escolhido para o projeto, como parte da atividade *Planejar Processo*. Na figura 7.31, o ADSOrg é instanciado.

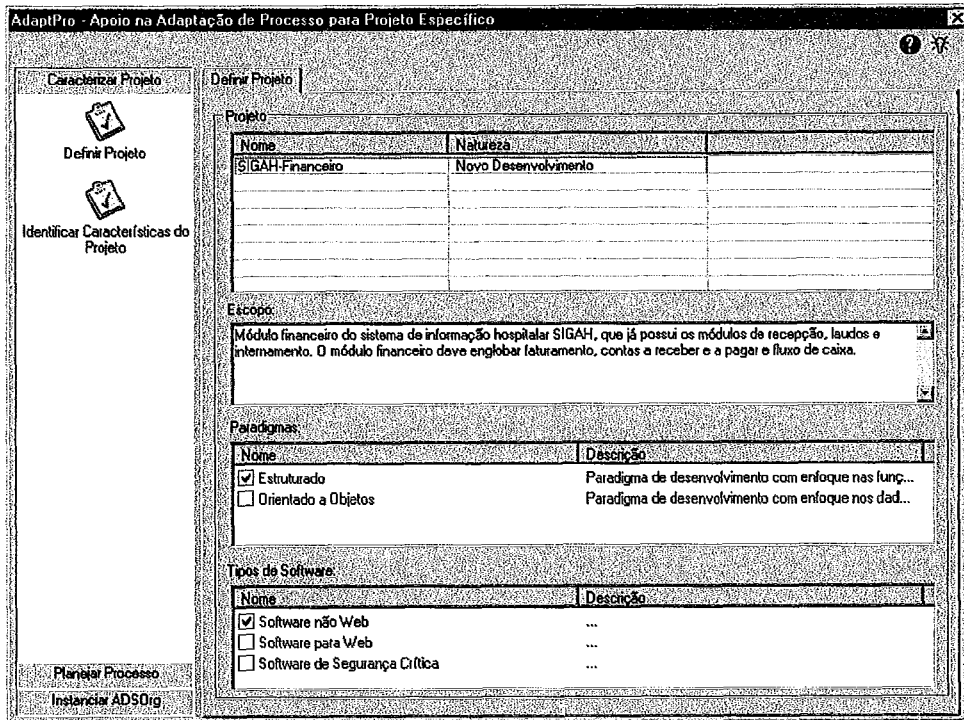


Figura 7.29 – Tela para Definição do Projeto

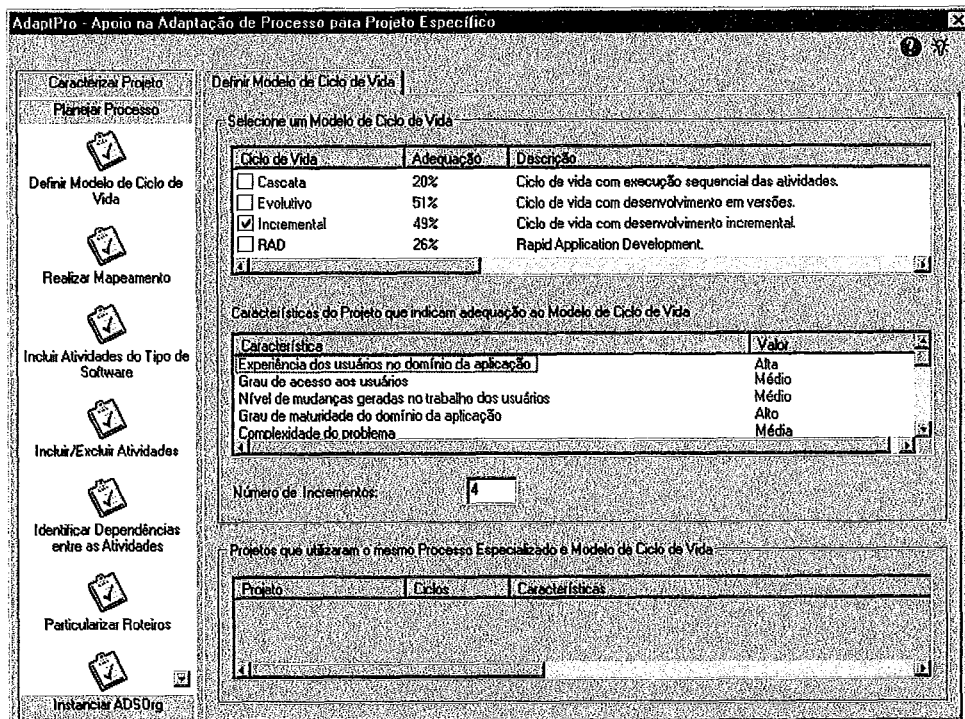


Figura 7.30 – Tela para Escolha do Modelo de Ciclo de Vida a ser Adotado no Projeto

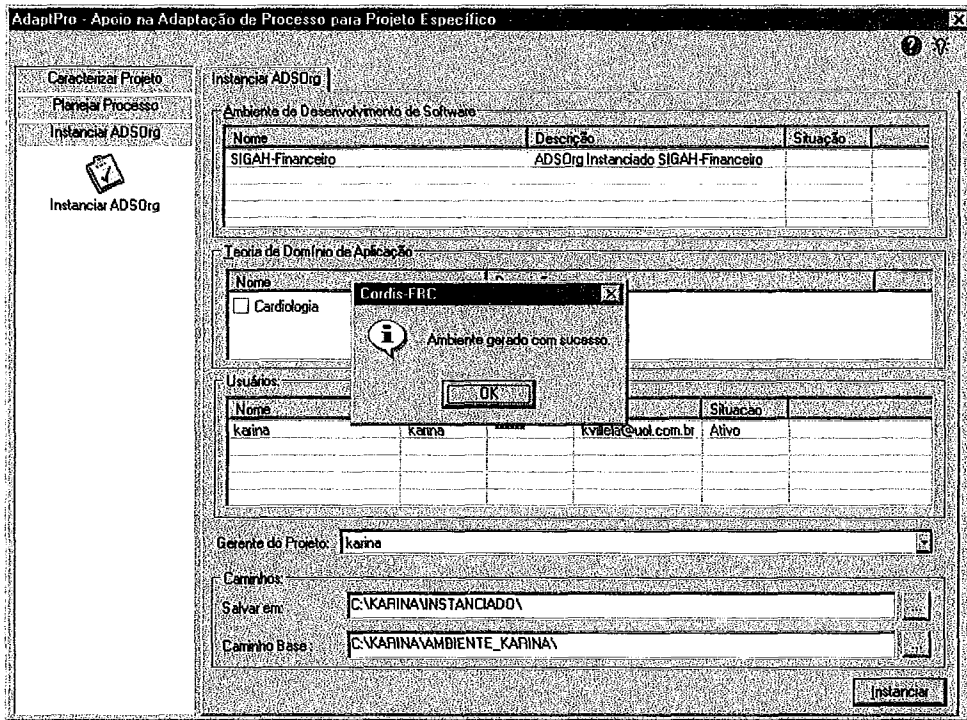


Figura 7.31 – Tela para Criação do ADSOrg Instanciado para o Projeto

Todas as ferramentas construídas no contexto do projeto “Ambientes de Desenvolvimento Orientados a Organização” para apoiar atividades dos processos de desenvolvimento e manutenção de software são orientadas a processo e possuem no lado direito superior da tela, dois ícones que, como mencionado anteriormente, têm a finalidade de fornecer o conhecimento disponível no ambiente para auxiliar a execução da atividade (🔍) e executar o módulo da ferramenta Acknowledge [31] que permite o registro de conhecimento obtido na execução da atividade (📝). Isto significa que as ferramentas possibilitam a gerência do conhecimento orientada a processo, ao fornecer conhecimento que é relevante para a atividade sendo executada e ao permitir o registro de conhecimento adquirido na execução da atividade. As figuras 7.32 e 7.33, por exemplo, ilustram como o gerente de projeto que precisa escolher um modelo de ciclo de vida para o projeto pode ter acesso ao conhecimento acumulado pela organização sobre esta atividade. Ao pressionar o botão do mouse sobre o ícone correspondente, os tipos de conhecimento disponíveis são listados (figura 7.32), uma busca pode ser realizada e o conhecimento desejado pode ser acessado (figura 7.33). No exemplo, a lição aprendida recuperada é sobre o uso do modelo de ciclo de vida evolutivo. O gerente do projeto pode também avaliar o grau de utilidade da lição aprendida (seção *Comentários do item de conhecimento* na figura 7.33), que será utilizado para apoiar a atividade de manutenção do conhecimento armazenado no repositório da organização.

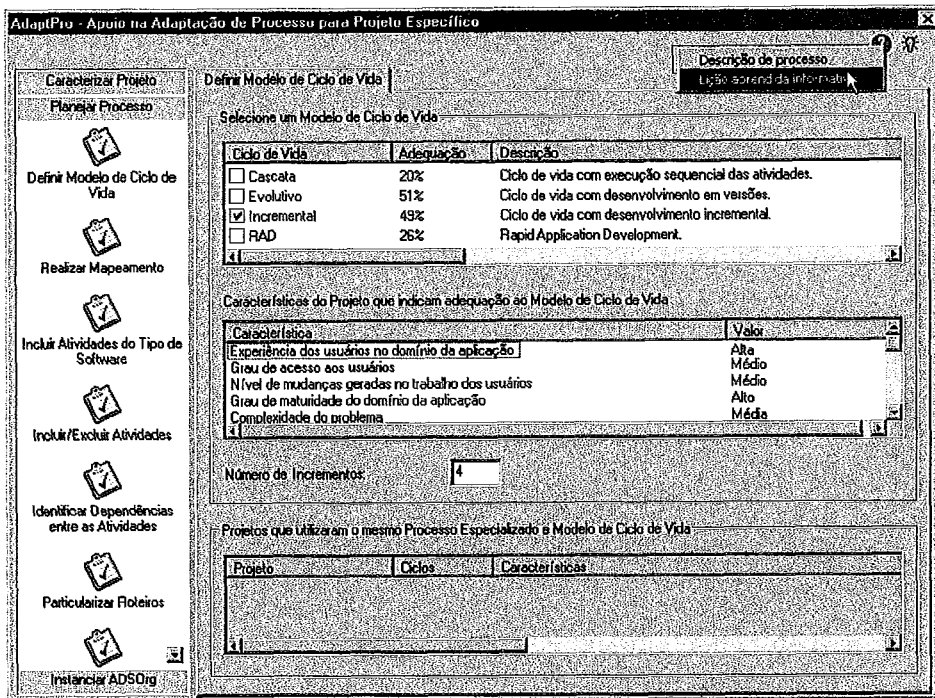


Figura 7.32 – Tela para Escolha do Modelo de Ciclo de Vida mostrando os Tipos de Conhecimento Disponíveis para Apoiar a Atividade

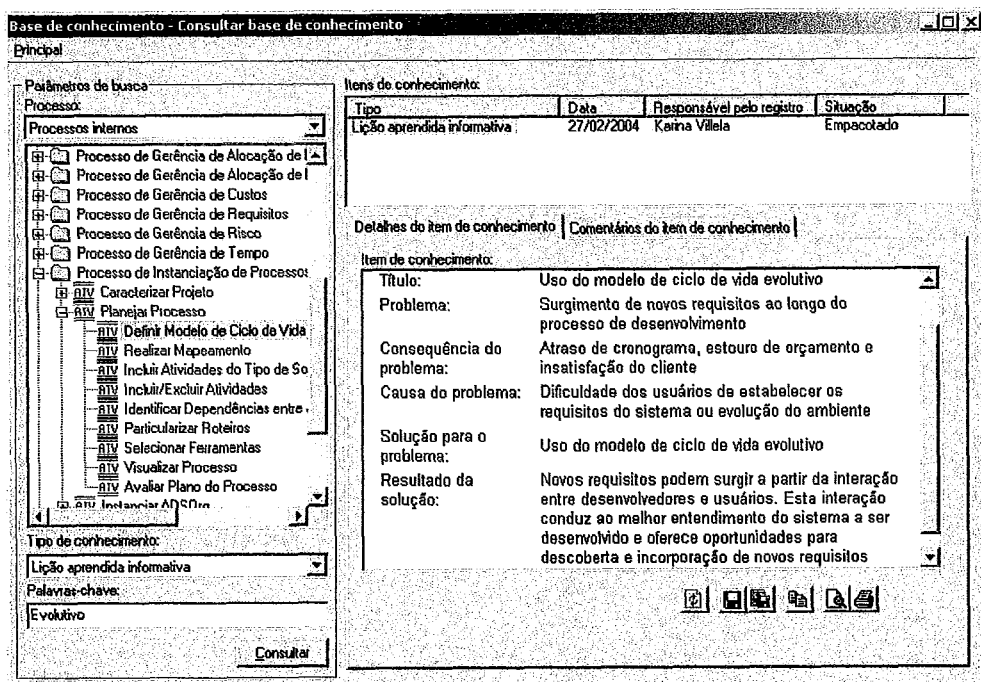


Figura 7.33 – Tela para Busca e Acesso do Conhecimento Desejado

A figura 7.34 apresenta a tela de um ADSOrg Instanciado a partir do Cordis-FBC, com as atividades do processo listadas no lado esquerdo. Quando uma atividade é selecionada para execução, as ferramentas de software que automatizam os procedimentos adotados para a execução da atividade são disponibilizadas no lado direito. Por exemplo, ao selecionar a atividade de planejamento de riscos, a ferramenta RiskManager é disponibilizada no lado direito, podendo, então, ser ativada.

Os artefatos a serem produzidos na atividade também são informados no lado direito da tela principal do ADSOrg Instanciado (figura 7.34), onde é fornecido acesso aos roteiros que devem ser utilizados. Além disso, esta figura ilustra que é possível obter e registrar conhecimento referente à atividade em execução, mesmo que não haja uma ferramenta de suporte à atividade, pois os ícones do lado direito superior estão sempre disponíveis. A figura 7.35 exibe a tela inicial da ferramenta RiscManager.

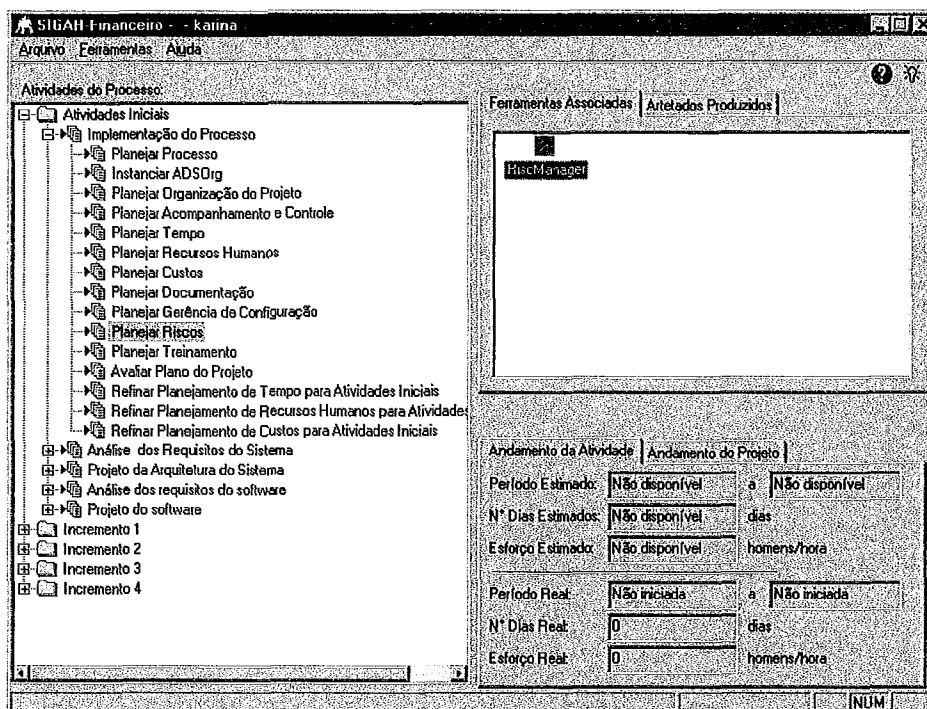


Figura 7.34 – Tela Principal do ADSOrg Instanciado

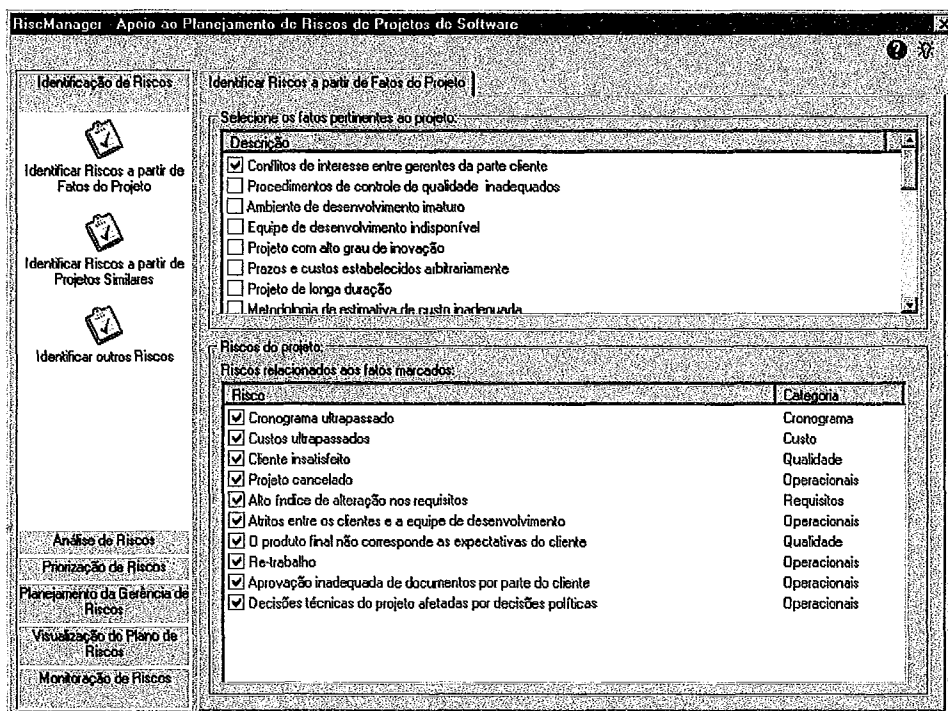


Figura 7.35 – Tela Inicial da RiscManager

Por fim, as figuras 7.36 e 7.37 mostram, respectivamente, um desenvolvedor de um projeto registrando uma idéia relacionada à atividade de Análise de Requisitos do Software no ADSOrg Instanciado e uma tela da ferramenta Acknowledge, na qual um conhecimento registrado está sendo encaminhado ao comitê de avaliação, de forma a a ser filtrado e, posteriormente, empacotado e disponibilizado.

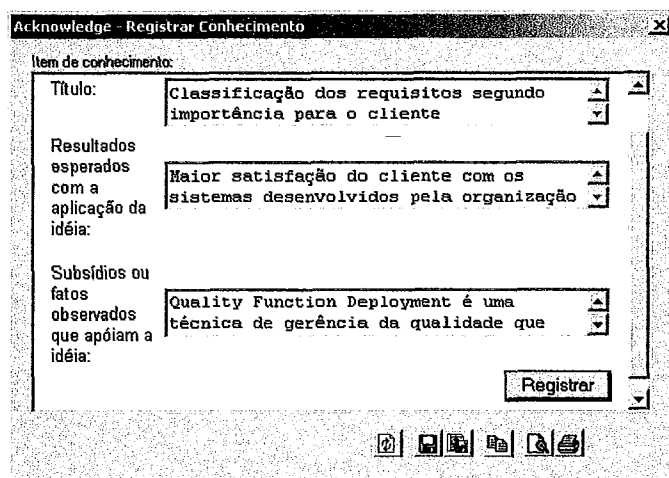


Figura 7.36 – Tela de Registro de Item de Conhecimento

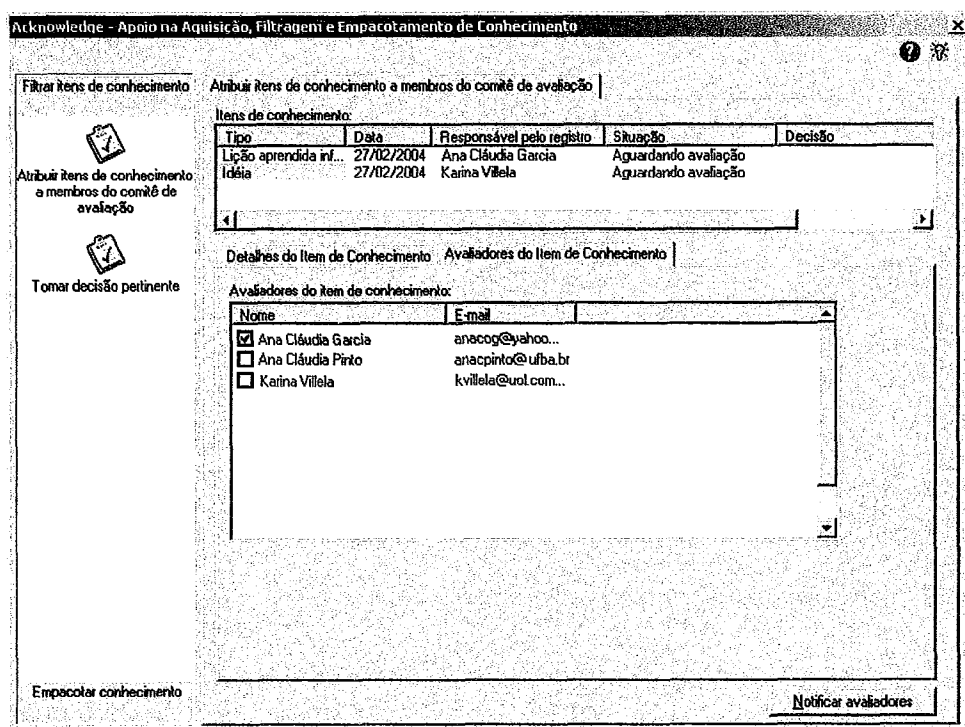


Figura 7.37 – Tela de Encaminhamento dos Itens de Conhecimento ao Comitê de Avaliação

A partir do uso do Cordis-FBC e dos ambientes a serem instanciados para projetos de software específicos, será possível avaliar o papel dos conhecimentos

acumulados sobre o domínio, sobre a organização e sobre a própria atividade de desenvolvimento dentro do contexto do desenvolvimento de software da organização.

Espera-se que os projetos de software da FBC sejam bastante beneficiados pelo uso dos ambientes propostos neste trabalho. Para o desenvolvimento de alguns tipos de software na FBC, o conhecimento sobre os processos organizacionais é muito importante, até mesmo essencial. Sistemas de informação hospitalar são um exemplo, pois acompanham e apóiam os processos de assistência médica desde a recepção do paciente até o faturamento da conta. Outro aspecto observado é a importância, principalmente durante as atividades de planejamento, de se identificar quem na organização detém o conhecimento necessário à execução de uma determinada atividade. Esta informação, relativamente simples de ser fornecida, pode assegurar que as pessoas mais adequadas à realização de uma tarefa sejam contatadas, ao mesmo tempo em que se evita o desperdício de tempo no processo de coleta de informações.

7.11 Conclusão

Este capítulo descreveu como a Estação TABA foi estendida para permitir a implementação do conceito de ADSOrg. A estratégia adotada prevê a criação de Ambientes Configurados para as organizações e de ADSOrg Instanciados para os projetos específicos de cada organização. Um Ambiente Configurado para uma organização, o *Cordis-FBC*, foi definido, gerado e preparado para o uso, demonstrando a viabilidade da abordagem.

Capítulo 8 – Conclusão e Perspectivas Futuras

Este capítulo conclui este trabalho, resumindo sua motivação e proposta, apresentando as suas contribuições e comparando a abordagem adotada com os sistemas e ambientes descritos nos capítulos de Gerência do Conhecimento e Ambientes de Desenvolvimento de Software. As perspectivas futuras fornecem a direção para que seja dada continuidade ao trabalho. Espera-se que os Ambientes de Desenvolvimento de Software Orientados a Organização (ADSOrg) não só apoiem os desenvolvedores de software, auxiliando na identificação do que funciona no contexto da organização, mas colaborem para a derivação de conclusões no nível de indústria e do domínio de conhecimento.

8.1. Conclusão

Processos de software são processos complexos, que necessitam de esforço coletivo e criativo e envolvem vários tipos de conhecimento. Ao mesmo tempo, os projetos precisam apresentar a melhor relação entre qualidade do produto, tempo para disponibilização do produto e custo, considerando os objetivos e características do projeto, o que torna fundamental a utilização eficiente do tempo. Para STATZ [7], isto significa fazer as coisas tão certas quanto possível, pois não há tempo para fazê-las novamente. Um requisito para ser capaz de fazer as coisas tão certas quanto possível, no contexto de atividades complexas e intensas em conhecimento como o desenvolvimento e a manutenção de software, é o de dispor ou ter acesso ao conhecimento necessário para a solução dos problemas que compõem a atividade a ser executada. Este é o foco dos ADSOrg, que, desta forma, pretendem complementar os esforços realizados em trabalhos anteriores, que buscaram fornecer suporte ao processo, apoio à adoção de procedimentos, integração de ferramentas e gerência dos artefatos produzidos ao longo do processo.

ADSOrg foram definidos, então, para apoiar a Gerência do Conhecimento ao longo dos processos de software. Vários tipos de conhecimento são importantes neste contexto e, muitas vezes, problemas ocorrem por que as equipes de projeto não dispõem de conhecimento suficiente sobre o domínio de aplicação, sobre a própria organização em que trabalham e/ou sobre as práticas de Engenharia de Software que apresentaram melhores resultados na organização. Em organizações em que o

desenvolvimento e a manutenção de software são atividades de negócio, conhecer o cliente é também essencial.

Sendo assim, um ADSOrg tem como objetivo fornecer o conjunto de conhecimentos relevantes para a execução das atividades de desenvolvimento e manutenção de software que tenham sido acumulados pela organização ao longo de seus vários projetos, além de apoiar o aprendizado organizacional em Engenharia de *Software* a partir do aprendizado nos projetos específicos.

De acordo com a definição fornecida dos componentes de um ADSOrg, o conhecimento é provido pela *Infra-estrutura de Gerência do Conhecimento*, que contém o *Repositório da Organização* e os *Serviços e Ferramentas de Gerência do Conhecimento*. O *Repositório da Organização* efetivamente contém o conhecimento acumulado pela organização para apoiar os desenvolvedores, sendo composto de: descrições de tarefas; *Teorias de Domínio*, dentre as quais se destaca a *Teoria do Domínio de ES (Teoria do Domínio de Engenharia de Software)*; descrição da organização a qual pertence o ambiente; descrição das organizações clientes, a depender do tipo de ADSOrg; e, principalmente, de bases que armazenam o conhecimento e os dados obtidos ao longo dos projetos de software e considerados relevantes para a organização. Os componentes de um ADSOrg também englobam *Repositórios de Projeto*, além de *Serviços e Ferramentas de Engenharia de Software*, que fazem uso da *Infra-estrutura de Gerência do Conhecimento* para fornecer conhecimento no momento em que ele se faz necessário para os desenvolvedores.

O trabalho de definição da *Infra-estrutura de Gerência do Conhecimento* contemplou a definição de uma ontologia de organização e de uma linguagem para modelagem de processo, ambas com o objetivo de permitir a descrição das organizações.

Além disso, verificou-se a necessidade de realização de uma pesquisa de campo para identificar quais conhecimentos priorizar quando definindo a *Teoria do Domínio de ES* e as bases de conhecimento e dados que compõem a *Infra-estrutura de Gerência do Conhecimento*. A partir dos resultados da pesquisa, trabalhos futuros poderão contemplar os tipos de conhecimento de acordo com as prioridades identificadas, definindo subteorias da *Teoria do Domínio de ES*, as estruturas necessárias para armazenamento dos respectivos dados e/ou conhecimento, além de ferramentas necessárias para manipular estes dados e/ou conhecimento. MONTONI [31] definiu estruturas genéricas para a representação de lições aprendidas, dúvidas, regras de associação, entre outros.

A definição de *Teorias de Domínio* para domínios de negócio específicos depende das organizações para as quais serão construídos os ADSOrg, fazendo

parte, portanto, do esforço de construção dos ambientes específicos. O componente *Descrição de Tarefas*, que já possui a tarefa de avaliação definida [137], apesar de ser genérico, independente de um domínio específico, também pode ter seu conteúdo definido de forma incremental, como parte dos esforços de construção de ambientes específicos, uma vez que a descrição de uma tarefa é útil na medida em que é aplicável a um domínio (conforme processo de configuração descrito no anexo I).

A Estação TABA, que foi desenvolvida e tem sido estendida no contexto do Projeto TABA, teve a sua infra-estrutura novamente ampliada para permitir a configuração de ambientes organizacionais capazes de instanciar ADSOrg para projetos específicos. Uma ferramenta de modelagem de processos foi desenvolvida para, junto com a ferramenta *Sapiens*, desenvolvida por SANTOS [25] para descrever as estruturas organizacionais, completarem a implementação básica da Infra-estrutura de Gerência do Conhecimento na Estação TABA.

Por fim, um Ambiente Configurado, denominado *Cordis-FBC*, foi gerado para a Fundação Bahiana de Cardiologia (FBC), organização citada no capítulo 1, a partir da infra-estrutura e do apoio fornecidos pela Estação TABA para a execução do processo de configuração. Em seguida, o ambiente foi preparado para uso, através do registro da estrutura organizacional, alocação de pessoal, distribuição de competências e processos organizacionais da FBC, possibilitando que conhecimento útil pudesse ser fornecido aos desenvolvedores desde o início da utilização dos Ambientes Configurado e Instanciados. Estes itens de conhecimento já haviam sido levantados, como parte deste trabalho, e utilizados para validação da ontologia de organização e da linguagem de modelagem de processos.

8.1.1. Contribuições

Conseqüentemente, as contribuições desta tese são:

- a definição de Ambientes de Desenvolvimento de Software Orientados a Organização (ADSOrg), o que implicou na:
 - definição dos requisitos para os ADSOrg;
 - definição de um modelo de componentes para a construção de ADSOrg;
 - construção de uma ontologia de organização, de forma a fornecer um vocabulário comum para a descrição das organizações;
 - definição de uma linguagem para a modelagem de processos, baseada na ontologia de organização, mas não restrita a mesma, e que permite a representação dos conhecimentos requeridos e produzidos ao longo dos processos;

- a identificação dos tipos de conhecimento considerados mais importantes pelos desenvolvedores de software no apoio as suas atividades, através da realização de uma pesquisa de campo na cidade de Salvador, que pode ser aplicada em outras cidades a partir dos questionários elaborados, sistema para aplicação de questionários desenvolvido e procedimentos de análise estatística utilizados;
- a extensão da Estação TABA para possibilitar a construção de ADSOrg, o que implicou na:
 - definição e implementação de uma estratégia para a construção de ADSOrg na Estação TABA, o que envolve a configuração do Meta-ambiente TABA para as organizações e instanciação de ADSOrg para os projetos específicos;
 - revisão das funções, ambientes, requisitos e arquitetura da Estação TABA, de forma a possibilitar a construção dos Ambientes Configurados e ADSOrg Instanciados;
 - definição de um processo para a configuração do meta-ambiente TABA para as organizações, que leva em consideração as particularidades e objetivos de cada organização, e de uma ferramenta que apóia este processo;
 - definição e construção de uma ferramenta para modelagem de processos baseada em um meta-modelo de dados, o que representa o suporte adequado à modelagem de processos, uma vez que a linguagem de modelagem utilizada pode evoluir de acordo com as necessidades observadas na prática;
 - apoio à definição e construção das ferramentas para descrição e visualização da estrutura organizacional [25]; para instanciação de processos de software para projetos específicos [30]; para apoio à captura de conhecimento ao longo dos processos de software [31]; para planejamento da documentação [37], riscos [27], tempo e custo [28]; e para planejamento, monitoração e avaliação da alocação de profissionais aos projetos de software [29];
- a configuração e preparação do *Cordis-FBC* para uso na Fundação Bahiana de Cardiologia, de forma a verificar, na prática, a viabilidade dos conceitos, modelos, estratégias e processos aqui propostos.

8.1.2. Comparação com Trabalhos Relacionados

Comparando a proposta de ADSOrg e o suporte efetivamente fornecido pelos Ambientes Configurados e ADSOrg Instanciados criados a partir da Estação TABA com os sistemas e ambientes de Gerência do Conhecimento apresentados nos capítulos 2 e 3, tem-se:

- a Infra-estrutura de Gerência do Conhecimento que compõe o modelo de componentes de um ADSOrg incorpora vários elementos utilizados em tais sistemas e ambientes, tais como: a ontologia de Engenharia de Software [74]; a ontologia de organização e a ontologia do domínio específico [46]; a descrição da estrutura e dos processos organizacionais, de forma a fornecer o contexto de criação e utilização dos itens de conhecimento, que é uma combinação das idéias de JABLONSKI *et al.* [90], NAGEL [17] e ABECKER *et al.* [16,46]; a utilização da descrição de processo como uma espécie de páginas amarelas, como proposto por DECKER e JEDLITSCHKA [85]; o mapa de competências, que é uma extensão do mapa de conhecimento proposto por ALTHOFF *et al.* [74];
- tais sistemas e ambientes não são ADS e, conseqüentemente, não provêm o suporte necessário à execução das atividades de um projeto de software. No entanto, alguns deles, assim como os ADSOrg, propõem a definição de processos de software para projetos específicos a partir de um processo padrão [17,86], chamado por HENNINGER [86] de metodologia de desenvolvimento padrão;
- a coleta de novos conhecimentos como proposta por WANGENHEIM *et al.* [58], ou seja, inserida, tanto quanto possível, na execução das atividades dos processos organizacionais, é a abordagem adotada pela ferramenta de apoio ao processo de captura de conhecimento desenvolvida por MONTONI [31] para uso na Estação TABA. O sistema descrito por NAGEL [17] também possui ferramentas para a coleta de conhecimento;
- assim como apresentado por ABECKER *et al.* [16], um ADSOrg fornece o conhecimento acumulado pela organização para um desenvolvedor a partir da informação sobre qual atividade ele está executando, restringindo, assim, o conteúdo fornecido, de forma a não sobrecarregá-lo desnecessariamente.

Já comparando os ADSOrg com os ADSOD, englobando o suporte fornecido pelos mesmos quando construídos a partir da Estação TABA, constata-se que os ADSOD contemplam o conhecimento sobre o domínio de aplicação, mas este conhecimento é apenas uma parcela do conhecimento necessário para os desenvolvedores de *software*. ADSOrg são uma extensão dos ADSOD que têm como objetivo prover, além do conhecimento sobre o domínio, conhecimento sobre Engenharia de Software, sobre as organizações envolvidas no projeto e sobre as práticas de engenharia de software que apresentaram melhores resultados na organização.

Por fim, na comparação com o sistema MILOS e o ambiente ODE, verifica-se que:

- todos são ADS Centrados em Processo que se baseiam na definição de modelos de processos para projetos específicos a partir de modelos de processos genéricos; prevêem a orientação do processo através da indicação das atividades a serem executadas, dos artefatos que são necessários para a execução das atividades e dos artefatos a serem produzidos;
- todos fornecem, aos desenvolvedores, os conhecimentos necessários à execução das atividades que lhes foram atribuídas no momento em que estes se tornam necessários e também permitem que os desenvolvedores registrem conhecimento que consideram que pode ser importante para outros desenvolvedores. No entanto, enquanto a gerência do conhecimento é orientada a processo no sistema MILOS e nos Ambientes Configurados e ADSOrg Instanciados, a gerência do conhecimento no ODE é baseada em agentes, sendo a distribuição do conhecimento um serviço provido pelas ferramentas do ambiente que apóiam atividades específicas. Esta característica do ODE não possibilita a disseminação do conhecimento associado às atividades ainda não apoiadas por ferramentas, a não ser que o usuário utilize o serviço de busca disponível no ambiente. Os Ambientes Configurados e ADSOrg Instanciados não apóiam a distribuição de conhecimento;
- o MILOS contempla itens de conhecimento acessíveis a partir de máquina de busca pública ou que estejam armazenados em bases de dados ou em um sistema de gerência de documento interno; o ODE contempla artefatos, instâncias de ontologias e lições aprendidas; os Ambientes Configurados e ADSOrg Instanciados contemplam roteiros de documentos, ontologias e suas instâncias, além de uma série de tipos de conhecimento previamente definidos (diretrizes, casos, lições aprendidas, dúvidas, idéias), permitindo, também, que novos tipos de itens de conhecimento sejam definidos e passem a ser capturados através da ferramenta *Acknowledge* [31];
- o sistema MILOS não prevê o auxílio ao entendimento do conhecimento que é fornecido ao desenvolvedor, que nos ADSOrg pode ser apoiado pela Teorias de Domínio e pelas descrições de tarefas. Assim como os ADSOrg, o ODE prevê o uso de ontologias de domínios de aplicação e de Engenharia de Software. Nos Ambientes Configurados e ADSOrg Instanciados, apenas a ontologia de processo de software compõe, atualmente, a Teoria de Engenharia de *Software*;
- apesar do sistema MILOS ter recentemente mencionado a intenção de incorporar modelos de habilidades¹ anexados às atividades e a membros da equipe de

¹ O conceito de habilidade é considerado de forma abrangente no contexto do sistema MILOS, equivalendo ao conceito de competência utilizado nos ADSOrg.

projeto, estes modelos ainda não foram propriamente definidos e implementados [135]. Nem o MILOS nem o ODE contemplam a descrição da organização que desenvolve e mantém software e, quando pertinente, de suas organizações clientes, o que representa conhecimento útil para a localização de especialistas, promoção da interação entre os profissionais da organização e levantamentos de requisitos relacionados aos processos organizacionais;

- tanto o ODE quanto os ADSOrg Instanciados a partir da Estação TABA permitem a avaliação, pelo usuário, dos itens de conhecimento utilizados e dispõem de ferramentas que apóiam atividades específicas dos processos de software, tais como as ferramentas para planejamento de riscos e de tempo e custo, capazes de fornecer conhecimento organizacional relacionado a estas atividades e, principalmente, de auxiliar a utilização do conhecimento fornecido;
- o ODE permite a configuração de agentes para apoiar na manutenção da memória organizacional e possui um serviço que informa quem utilizou um determinado conhecimento, além de quando e quão freqüentemente ele tem sido utilizado, o que ainda não está disponível nos Ambientes Configurados e ADSOrg Instanciados e também não é mencionado nas descrições do sistema MILOS;
- o sistema MILOS contempla alguns requisitos que foram definidos no contexto da Estação TABA, mas que ainda não foram contemplados, os quais são: suporte à modificação dos modelos de processos utilizados nos projetos específicos e apoio ao empacotamento de bons elementos desses modelos em modelos de processos reutilizáveis. O MILOS também realiza estimativas de conclusão de atividade, verificando a possível violação do cronograma do projeto e, se necessário, alertando o gerente de projeto.

Em resumo, um ADSOrg integra várias características e funcionalidades de sistemas e ambientes de Gerência do Conhecimento em um ADS Centrado em Processo, destacando-se, em relação ao MILOS, por promover o entendimento do conhecimento organizacional disponível; fornecer conhecimento sobre a organização que desenvolve e mantém software e, quando pertinente, sobre a organização cliente; e possuir ferramentas que apóiam atividades específicas dos projetos de software, não só fornecendo o conhecimento organizacional necessário à execução da atividade, mas também auxiliando a utilização deste conhecimento na execução propriamente dita da atividade. Em relação ao ODE, os ADSOrg destacam-se por fornecer conhecimento sobre as organizações, conforme já mencionado, e por apoiar a gerência do conhecimento orientada a processo, podendo, em trabalhos futuros,

explorar características interessantes do ambiente ODE, como o uso da tecnologia de agentes e a ontologia de Qualidade de Software já construída [170].

8.2. Perspectivas Futuras

Muito trabalho foi realizado no contexto desta tese no sentido de viabilizar o uso de ADSOrg para apoiar efetivamente o desenvolvimento, reparo e melhoria de produtos de software. No entanto, ainda existe bastante trabalho a ser realizado:

- extensão da ferramenta AdaptPro [30] para que a descrição do processo para o projeto seja gerada em XML, de forma a ser utilizada pela máquina de processo construída por MAFRA [36].
- extensão da ferramenta EDITAR [137] para permitir a localização das referências a uma determinada tarefa no repositório da organização e implementação de um serviço que permita a visualização do conhecimento referente a uma tarefa a partir de outras ferramentas do ambiente;
- a definição da *Teoria do Domínio de ES* a partir dos resultados da pesquisa de campo, de forma a fornecer um vocabulário comum para descrever o conhecimento sobre Engenharia de Software armazenado nos repositórios da organização e dos projetos e auxiliar o melhor entendimento do mesmo;
- extensão da ferramenta NAVEGUE [148] para permitir a localização das referências a um determinado conceito ou instância de conceito no repositório da organização e implementação de um serviço que permita a visualização do conhecimento referente a um conceito ou uma instância de um conceito a partir de outras ferramentas do ambiente, o que inclui menção aos conceitos ou instâncias de conceitos relacionados;
- definição e construção de uma ferramenta para recuperação e acesso a itens de conhecimento armazenados no repositório da organização, a partir dos conceitos e relações das ontologias descritas no ambiente e, também, das instâncias desses conceitos e relações;
- definição e construção de serviço/ferramenta para análise do uso dos itens de conhecimento fornecidos no *Repositório da Organização* de acordo com a frequência de uso, grau de utilidade e comentários gerais dos desenvolvedores, de forma a identificar itens de conhecimento candidatos à evolução ou à exclusão;
- definição e construção de um serviço/ferramenta para identificação de itens de conhecimento similares no *Repositório da Organização*, de forma oferecer oportunidades para consolidação dos mesmos;

- investigação da necessidade de funcionalidades específicas em Ambientes Configurados para Organizações de Software e em seus ADSOrg Instanciados e fornecimento da possibilidade de especialização do processo padrão da organização para um determinado cliente;
- definição e construção de ferramentas que apoiem atividades específicas dos projetos de software, ao fornecer e auxiliar a utilização do conhecimento organizacional necessário à execução das mesmas. Isto deve ser feito de acordo com as prioridades identificadas na pesquisa de campo. A definição das ferramentas deve incluir a definição da estrutura utilizada para armazenamento do conhecimento organizacional a respeito do tópico abordado e a definição da porção da *Teoria do Domínio de ES* que descreve o conhecimento relativo ao tópico;
- definição e construção de um serviço que permita a configuração de um novo ambiente para uma organização (nova versão do Ambiente Configurado), sem acarretar a perda do conhecimento acumulado na versão anterior;
- avaliação de experiências de uso dos Ambientes Configurados e ADSOrg Instanciados na indústria. Este trabalho já está sendo iniciado em 8 empresas de Salvador, através de um projeto conjunto da ASSESPRO-BA, COPPE/UFRJ e UNIFACS.

Apesar da pesquisa de campo ter sido realizada apenas na cidade de Salvador, os seus resultados fornecem indicações importantes de quais tópicos precisam ser abordados por futuros trabalhos de pesquisa, de forma a suprir as necessidades de conhecimento dos desenvolvedores de software.

A partir das experiências de uso dos Ambientes Configurados e ADSOrg Instanciados na indústria será possível avaliar a eficácia da abordagem proposta neste trabalho.

Referências Bibliográficas

- [1] DESOUZA, K. C., "Barriers to Effective Use of Knowledge Management Systems in Software Engineering, *Communications of ACM*, v. 46, n. 1, pp. 99-101, Jan. 2003.
- [2] WATERSON, P. E., CLEGG, C. W., AXTELL, C. M., "The dynamics of work organization, knowledge and technology during software development", *International Journal of Human-Computer Studies*, v. 46, n. 1, pp. 79-101, Jan. 1997.
- [3] LINDVALL, M., FREY, M., COSTA, P., *et al.*, "Lessons Learned about Structuring and Describing Experience for Three Experience Bases". In: *Advances in Learning Software Organizations*, v. 2176, *Lecture Notes in Computer Science*, Springer, pp. 106-119, 2001.
- [4] JURISTO, N., MORENO, A. M., "Reliable Knowledge for Software Development", *IEEE Software*, v. 19, n. 5, pp. 98-99, Sep/Oct. 2002.
- [5] OH, E., HOEK, A., "Adapting Game Technology to Support Individual and Organizational Learning". In: *Proceedings of 13th International Conference on Software Engineering & Knowledge Engineering*, pp. 347-362, Buenos Aires, Argentina, Jun. 2001.
- [6] MAURER, F., HOLZ, H., "Process-centered Knowledge Organization for Software Engineering". In: Aha, D.W., Avila, H.M. (eds), *Exploring Synergies of Knowledge Management and Case-Based Reasoning: A 1999 AAAI Workshop*, Technical Report AIC-99-008, Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence, Washington, USA, 1999.
- [7] STATZ, J., "Leverage Your Lessons", *IEEE Software*, v. 16, n. 2, pp. 30-32, Mar/Apr. 1999.
- [8] RUS, I., LINDVALL, M., "Knowledge Management in Software Engineering", *IEEE Software*, v. 19, n. 3, pp. 26-38, May/Jun. 2002.
- [9] BIRK, A., KRÖSCHEL, F., "A Knowledge Management Lifecycle for Experience Packages on Software Engineering Technologies". In: *Proceedings of the Workshop on Learning Software Organizations*, pp. 115-126, Kaiserslautern, Germany, Jun. 1999.
- [10] ALLWEYER, T., "A Framework for Re-designing and Managing Knowledge Processes", ARIS White Paper, IDS Scheer AG, Saarbrücken, Germany, 1999. In: <http://www.changeware.net/doc/km.pdf> (acessado em 02/03/2004).

- [11] MURCH, R., "Knowledge Management", *Project Management: Best Practices for IT Professionals*, 1st edition, chapter 16, Prentice Hall, 2000.
- [12] KUCZA, T., NÄTTINEN, M., PARVIAINEN, P., "Improving Knowledge Management in Software Reuse Process". In: *Proceedings of Third International Conference on Product Focused Software Process Improvement*, v. 2188, *Lecture Notes in Computer Science*, Springer-Verlag, pp. 141-152, 2001.
- [13] MEEHAN, B., RICHARDSON, I., "Identification of Software Process Knowledge Management", *Software Process Improvement and Practice*, v. 7, n. 2, pp. 47-55, Jun. 2002.
- [14] PRAHALAD, C. K., HAMEL, G., "The Core Competence of the Corporation", *Harvard Business Review*, v. 38, n. 3, pp. 79-91, May/June. 1990.
- [15] PFLEEGER, S., "Albert Einstein and Empirical Software Engineering", *IEEE Computer*, v. 32, n. 10, pp. 32-38, Oct. 1999.
- [16] ABECKER, A., MENTZAS, G., LEGAL, M., *et al.*, "Business-Process Oriented Delivery of Knowledge through Domain Ontologies". In: *Proceedings of 12th International Workshop on Database and Expert Systems Applications*, pp. 442-446, Munich, Germany, Sep. 2001.
- [17] NAGEL, C., "Process and Knowledge Management: A Symbiosis". In: *Proceedings of Third International Conference on Product Focused Software Process Improvement*, v. 2188, *Lecture Notes in Computer Science*, Springer-Verlag, pp. 153-166, 2001.
- [18] MAIER, R., REMUS, U., "Towards a Framework for Knowledge Management Strategies: Process Orientation as Strategic Starting Point". In: *Proceedings of the 34th Hawaii International Conference on System Sciences*, pp. 4023, Maui, Hawaii, Jan. 2001.
- [19] O'LEARY, D. E., STUDER, R., "Knowledge Management: An Interdisciplinary Approach", *IEEE Intelligent Systems*, v. 16, n. 1, pp. 24-25, Jan/Feb. 2001
- [20] OLIVEIRA, K., *Modelo para Construção de Ambientes de Desenvolvimento de Software Orientados a Domínio*, Tese de D. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 1999.
- [21] VILLELA, K., TRAVASSOS, G. H., ROCHA, A. R., "Toward Enterprise Oriented Software Development Environments". In: *Workshop de Teses e Dissertações em Engenharia de Software - XIV Simpósio Brasileiro de Engenharia de Software*, pp. 379-384, João Pessoa, Brasil, Out. 2000.

- [22] ROCHA, A. R., SOUZA, J. M., AGUIAR, T. C., "TABA: A Heuristic Workstation for Software Development". In: *Proceedings of COMPEURO 90*, pp. 126-129, Tel Aviv, Israel, May 1990.
- [23] TRAVASSOS, G. H., *O Modelo de Integração de Ferramentas da Estação TABA*, Tese de D. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 1994.
- [24] MEGGINSON, L., MOSLEY, D., PIETRI, P., "Fundamentos do Modelo Organizacional", *Administração: Conceitos e Aplicações*, capítulo 8, Editora Harbra, 1986b.
- [25] SANTOS, G., *Representação da Distribuição do Conhecimento, Habilidades e Experiências através da Estrutura Organizacional*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2003.
- [26] BONFIM, C., *Uma Ferramenta para Modelagem de Processos em Ambientes de Desenvolvimento de Software Orientados à Organização*, Monografia de Final de Curso de B.Sc., UFBA, Salvador, Brasil, 2001.
- [27] FARIAS, L., *Planejamento de Riscos em Ambientes de Desenvolvimento de Software Orientados à Organização*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2002.
- [28] BARCELLOS, M., *Planejamento de Custos em Ambientes de Desenvolvimento, de Software Orientados à Organização*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2003.
- [29] SCHNAIDER, L., *Planejamento da Alocação de Recursos Humanos em Ambientes de Desenvolvimento, de Software Orientados à Organização*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2003.
- [30] BERGER, P., *Instanciação de Processos de Software em Ambientes Configurados na Estação TABA*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2003.
- [31] MONTONI, M., *Aquisição de Conhecimento: Uma Aplicação no Processo de Software*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2003.
- [32] MONTONI, M. A., *CardioKnowledge: Um Ambiente de Gerência do Conhecimento para Cardiologia*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2003.
- [33] COSTA, V., *Tecknowledge: Um Ambiente de Gerência do Conhecimento para uma Organização Fornecedora de Software*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2003.
- [34] AMARAL, E., *Empacotamento de Experimentos em Engenharia de Software*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2003.

- [35] FIGUEIREDO, S., *Gerência de Configuração em Ambientes de Desenvolvimento de Software Orientados à Organização*, Monografia de Final de Curso de B.Sc., UFRJ, Rio de Janeiro, Brasil, 2004.
- [36] MAFRA, S., *Infra-estrutura para Automatização de Processos de Software*, Monografia de Final de Curso de B.Sc., UFRJ, Rio de Janeiro, Brasil, 2004.
- [37] MARTINS, F., *Ambientes de Desenvolvimento de Software Orientado à Organização Baseado em Instrumentação Virtual*, Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2004.
- [38] RUBENSTEIN-MONTANO, B., LIEBOWITZ, J., BUCHWALTER, J., *et al.*, "A Systems Thinking Framework for Knowledge Management, *Decision Support Systems*, v. 31, n. 1, pp. 5-16, May 2001.
- [39] MARKKULA, M., "Knowledge Management in Software Engineering Projects". In: *Proceedings of the 11th International Conference on Software Engineering & Knowledge Engineering*, pp. 20-27, Kaiserslautern, Germany, Jun. 1999.
- [40] DAVENPORT, T., PRUSAK, L., *Working Knowledge: How Organizations Manage What They Know*, Boston, USA, Harvard Business School Press, 1998. Ref: MARKKULA (1999).
- [41] WIEDERHOLD, G., "Knowledge versus Data". In: Michael L. Brodie e John Mylopoulos (eds.), *On Knowledge Base Management Systems*, chapter 7, New York, USA, Springer-Verlag, 1986.
- [42] NONAKA, I., TAKEUCHI, H., *The Knowledge-Creation Company: How Japanese Companies Create the Dynamics of Innovation*, Oxford, England, Oxford University Press, 1995. Ref: MARKKULA (1999), SKYRME (1998).
- [43] MACINTOSH, A., "Position Paper on Knowledge Asset Management". In: <http://www.ntgi.net/ntgi/y2k/info/kmfr.htm> [Information and Research], 1997. Ref: MEEHAN e RICHARDSON (2002).
- [44] DING, H., RAVICHANDRAN, T., "Pre-emptive Radical Innovation: Building Inter-departmental Common Knowledge in a Short Product Development Cycle". In: *Proceedings of the 2000 IEEE Engineering Management Society*, pp. 575-580, Albuquerque, New Mexico, Aug. 2000.
- [45] NONAKA, I., "The Knowledge-Creating Company", *Harvard Business Review*, v. 69, n. 6, pp. 96-104, 1991.
- [46] ABECKER, A., BERNADI, A., HINKELMANN, K., *et al.*, "Toward a Technology for Organizational Memories", *IEEE Intelligent Systems*, v. 13, n. 3, pp. 40-48, May/Jun. 1998.

- [47] KEMP, L. L., NIDIFFER, K. E., ROSE, L. C., *et al.*, "Knowledge Management: Insights from the Trenches", *IEEE Software*, v. 18, n. 6, pp. 66-68, Nov/Dec. 2001.
- [48] LIEBOWITZ, J., "A Look at Nasa Goddard Space Flight Center's Knowledge Management Initiatives", *IEEE Software*, v. 19, n. 3, pp. 40-42, May/June. 2002.
- [49] WEI, C., HU, P., CHEN, H., "Design and Evaluation of a Knowledge Management System", *IEEE Software*, v. 19, n. 3, pp. 56-59, May/June. 2002.
- [50] HENDRIKS, P., VRIENS, D., "Knowledge-based systems and knowledge management: Friends or foes?", *Information & Management*, v. 35, n. 2, pp. 113-125, Feb. 1999.
- [51] BECERRA-FERNANDEZ, I., "Center for Innovation and Knowledge Management", *SIGGROUP Bulletin*, v. 19, n. 1, pp. 46-51, Apr. 1998.
- [52] JONES, M., "Knowledge Management at Work: Current Approaches and Case Studies", *SIGGROUP Bulletin*, v. 19, n. 1, pp. 33-34, Apr. 1998.
- [53] KOUWENHOVEN, T., "Reengineering for Learning", *SIGGROUP Bulletin*, v. 19, n. 1, pp. 39-45, Apr. 1998.
- [54] O'LEARY, D. E., "Using AI in Knowledge Management: Knowledge Bases and Ontologies", *IEEE Intelligent Systems*, v. 13, n. 3, pp. 34-39, May/June. 1998.
- [55] WHITEHALL, B., "AAAI-97 Workshop on AI and Knowledge Management", *AI Magazine*, v. 19, n. 1, pp. 130, Spring 1998.
- [56] FRANK, U., "Multi-Perspective Enterprise Models as a Conceptual Foundation for Knowledge Management". In: *Proceedings of the 33rd Hawaii International Conference on System Sciences*, pp. 3021, Maui, Hawaii, Jan. 2000.
- [57] DIENG, R., "Knowledge Management and the Internet", *IEEE Intelligent Systems*, v. 15, n. 13, pp. 14-17, May/June. 2000.
- [58] WANGENHEIM, C. G., LICHTNOW, D., WANGENHEIM, A., "A Hybrid Approach for Corporate Memory Management Systems in Software R&D Organizations". In: *Proceedings of 13th International Conference on Software Engineering & Knowledge Engineering*, pp. 326-330, Buenos Aires, Argentina, June. 2001.
- [59] PREECE, A., FLETT, A., SLEEMAN, D., *et al.*, "Better Knowledge Management through Knowledge Engineering", *IEEE Intelligent Systems*, v. 16, n. 1, pp. 36-43, Jan/Feb. 2001.
- [60] LIEBOWITZ, J., BECKMAN, T., *Knowledge Organizations: What Every Manager Should Know*, Boca Raton, USA, St. Lucie/CRC Press, 1998.
- [61] TIWANA, A., "Custom KM: Implementing the Right Knowledge Management Strategy for Your Organization", *Cutter IT Journal*, v. 12, n. 11, pp. 6-14, Nov. 1999.

- [62] WANGENHEIM, C. G., LICHTNOW, D., WANGENHEIM, A., *et al.*, "Supporting Knowledge Management in University Software R&D Groups". In: *Advances in Learning Software Organizations*, v. 2176, *Lecture Notes in Computer Science*, Springer, pp. 52-66, 2001b.
- [63] SKYRME, D., "Knowledge Management Solutions - The IT Contribution", *SIGGROUP Bulletin*, v. 19, n. 1, pp. 34-39, Apr. 1998.
- [64] LUCCA, J., SHARDA, R, WEISER, M., "Coordinating Technologies for Knowledge Management in Virtual Organizations". In: *Academia/Industry Working Conference on Research Challenges*, pp. 21-26, Buffalo, USA, Apr. 2000.
- [65] NICK, M., ALTHOFF, K., "Engineering Experience Base Maintenance Knowledge". In: *Advances in Learning Software Organizations*, v. 2176, *Lecture Notes in Computer Science*, Springer, pp. 222-236, 2001.
- [66] BASILI, V., LINDVALL, M., COSTA, C., "Implementing the Experience Factory Concepts as a set of Experiences Bases". In: *Proceedings of 13th International Conference on Software Engineering & Knowledge Engineering*, pp. 102-109, Buenos Aires, Argentina, Jun. 2001.
- [67] TRITTMANN, R., "The Organic and the Mechanistic Form of Managing Knowledge in Software Development". In: *Advances in Learning Software Organizations*, v. 2176, *Lecture Notes in Computer Science*, Springer, pp. 22-36, 2001.
- [68] PROBST, G., RAUB, S., ROMHARDT, K., *Managing Knowledge: Building Blocks for Success*, Chichester, England, John Wiley & Sons, 2000.
- [69] SNOEK, B., *Knowledge Management and Organizational Learning: Systematic Development of an Experience Base on Approaches and Technologies*, Diploma Thesis, Fraunhofer Institute for Experimental Software Engineering, Kaiserslautern, Germany, 1999.
- [70] FELDMANN, R., ALTHOFF, K., "On the Status of Learning Software Organizations in the Year 2001". In: *Advances in Learning Software Organizations*, v. 2176, *Lecture Notes in Computer Science*, Springer, pp. 2-6, 2001.
- [71] O'LEARY, D., "How Knowledge Reuse Informs Effective System Design and Implementation", *IEEE Intelligent Systems*, v. 16, n. 1, pp. 44-49, Jan/Feb. 2001.
- [72] BASILI, V., "The Experience Factory: Packaging Software Experience". In: *Proceedings of the 14th Annual Software Engineering Workshop*, Software Engineering Laboratory, Goddard Space Flight Center - NASA, Greenbelt, USA, Nov. 1989. Ref: HOUDEK e SCHNEIDER (1999).
- [73] MENDONÇA, M. G., SEAMAN, C. B., BASILI, V., *et al.*, "A Prototype Experience Management System for a Software Consulting Organization" . In:

Proceedings of 13th International Conference on Software Engineering & Knowledge Engineering, pp. 29-36, Buenos Aires, Argentina, Jun. 2001.

- [74] ALTHOFF, K., BIRK, A., HARTKOPF, S. *et al.*, "Managing Software Engineering Experience for Comprehensive Reuse". In: *Proceedings of the 11th International Conference on Software Engineering & Knowledge Engineering*, pp. 10-19, Kaiserslautern, Germany, Jun. 1999.
- [75] HOUDEK, F., SCHNEIDER, K., "Software Experience Center: The Evolution of the Experience Factory Concept". In: *Proceedings of the 24th Annual Software Engineering Workshop*, Software Engineering Laboratory, Goddard Space Flight Center - NASA, Greenbelt, USA, Dec. 1999.
- [76] KOENNECKER, A., JEFFERY, R., LOW, G., "Lessons Learned from the Failure of an Experience Base Initiative Using a Bottom-up Development Paradigm". In: *Proceedings of the 24th Annual Software Engineering Workshop*, Software Engineering Laboratory, Goddard Space Flight Center - NASA, Greenbelt, USA, Dec. 1999.
- [77] SCHNEIDER, K., HUNNIUS, J., BASILI, V., "Experience in Implementing a Learning Software Organization", *IEEE Software*, v. 19, n. 3, pp. 46-49, May/June. 2002.
- [78] TAUTZ, C., *Customizing Software Engineering Experience Management Systems to Organisational Needs*, Dissertation am Fachbereich Informatik der Universität, Kaiserslautern, Germany, 2000.
- [79] ALAVI, M., LEIDNER, D. E., "Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues", *MIS Quarterly*, v. 25, n. 1, pp. 107-136, Mar. 2001. Ref: RAMESH (2002).
- [80] STAAB, S., STUDER, R., SCHNURR, H., *et al.*, "Knowledge Process and Ontologies", *IEEE Intelligent Systems*, v. 16, n. 1, pp. 26-34, Jan/Feb. 2001.
- [81] SCHNEIDER, K., SCHWINN, T., "Maturing Experience Base Concepts at DaimlerChrysler", *Software Process Improvement and Practice*, v. 6, n. 2, pp. 85-96, Jun. 2001.
- [82] ABECKER, A., BERNADI, A., SINTEK, M., "Proactive Knowledge Delivery For Enterprise Knowledge Management". In: *Proceedings of the 11th International Conference on Software Engineering & Knowledge Engineering*, pp. 120-127, Kaiserslautern, Germany, Jun. 1999.
- [83] ACKERMAN, M., HALVERSON, C., "Reexamining Organizational Memory", *Communications of the ACM*, v. 43, n. 1, pp. 59-64, Jan. 2000.

- [84] SCHNEIDER, K., HUNNIUS, J., "Effective Experience Repositories for Software Engineering". In: *Proceedings of the 25th International Conference on Software Engineering (ICSE'03)*, pp. 534-539, Portland, USA, May. 2003.
- [85] DECKER, B., JEDLITSCHKA, A., "The Integrated Corporate Information Network iCoIN: A Comprehensive, Web-Based Experience Factory". In: *Advances in Learning Software Organizations*, v. 2176, *Lecture Notes in Computer Science*, Springer, pp. 192-206, 2001.
- [86] HENNINGER, S., "Turning Development Standards into Repositories of Experiences", *Software Process Improvement and Practice*, v. 6, n. 3, pp. 141-155, Sep. 2001.
- [87] FALBO, R., NATALI, A. C., MIAN, P., *et al.*, "ODE – Ontology-based software Development Environment". In: *IX Congreso Argentino de Ciencias de la Computación*, pp. 1124-1135, La Plata, Argentina, Oct. 2003.
- [88] RAMASUBRAMANIAN, S, JAGADEESAN, G., "Knowledge Management at Infosys", *IEEE Software*, v. 19, n. 3, pp. 53-55, May/June. 2002.
- [89] WANGENHEIM, C. G., RODRIGUES, M. R., "Planejamento de Programas de Mensuração Baseado em Reutilização". In: *XI CITS – Qualidade de Software*, pp. 44-57, Curitiba, Brasil, Jun. 2000.
- [90] JABLONSKI, S., HORN, S., SCHLUNDT, M., "Process Oriented Knowledge Management". In: *Proceedings of the 11th International Workshop on Research Issues in Data Engineering*, pp. 77-84, Heidelberg, Germany, Apr. 2001.
- [91] RAMESH, B., "Process Knowledge Management with Traceability", *IEEE Software*, v. 19, n. 3, pp. 50-52, May/June. 2002.
- [92] O'LEARY, D. E., "Enterprise Knowledge Management", *IEEE Computer*, v. 31, n. 3, pp. 54-61, Mar. 1998.
- [93] RABARIJAONA, A., DIENG, R., CORBY, O., *et al.*, "Building and Searching an XML-Based Corporate Memory", *IEEE Intelligent Systems*, v. 15, n. 13, pp. 56-63, May/June. 2000.
- [94] HOLZ, H., KÖNNECKER, A., MAURER, F., "Task-Specific Knowledge Management in a Process-Centred SEE". In: *Advances in Learning Software Organizations*, v. 2176, *Lecture Notes in Computer Science*, Springer, pp. 163-177, 2001.
- [95] USCHOLD, M., GRUNINGER, M., "Ontologies: principles, methods and applications", *The Knowledge Engineering Review*, v. 11, n. 2, pp 93-136, 1996.

- [96] HENNINGER, S., "Keynote Address: Organizational Learning in Dynamic Domains". In: *Advances in Learning Software Organizations*, v. 2176, *Lecture Notes in Computer Science*, Springer, pp. 8-16, 2001.
- [97] PRIETRO-DÍAZ, R., *A Software Classification Scheme*, Ph.D. Thesis, University of California-Irvine, Irvine, USA, 1985.
- [98] KAPPE, F., "Managing Knowledge with Hyperwave Information Server", *Hyperwave White Paper*, version 1.2, 1999.
- [99] DINGSOYR, T., ROYRVIK, E., "Skills Management as Knowledge Technology in a Software Consultancy Company". In: *Advances in Learning Software Organizations*, v. 2176, *Lecture Notes in Computer Science*, Springer, pp. 96-103, 2001.
- [100] SCHNEIDER, K., "Experience Magnets: Attracting Experiences, Not Just Storing Them". In: *Proceedings of Third International Conference on Product Focused Software Process Improvement*, v. 2188, *Lecture Notes in Computer Science*, Springer-Verlag, pp. 126-140, 2001.
- [101] DINGSOYR, T., ROYRVIK, E., "An Empirical Study of an Informal Knowledge Repository in a Medium-Sized Software Consulting Company". In: *Proceedings of 25th International Conference on Software Engineering*, pp. 84-92, Portland, USA, May 2003.
- [102] KOMI-SIRVIÖ, S., MÄNTYNIEMI, A., SEPPÄNEN, V., "Toward a Practical Solution for Capturing Knowledge for Software Projects", *IEEE Software*, v. 19, n. 3, pp. 60-62, May/Jun. 2002.
- [103] DINGSOYR, T., MOE, N., NYTRO, O., "Augmenting Experience Reports with Lightweight Postmortem Reviews". In: *Proceedings of Third International Conference on Product Focused Software Process Improvement*, v. 2188, *Lecture Notes in Computer Science*, Springer-Verlag, pp. 167-181, 2001.
- [104] BIRK, A., DINGSOYR, T., STALHANE, T., "Postmortem: Never Leave a Project without It", *IEEE Software*, v. 19, n. 3, pp. 43-45, May/Jun. 2002.
- [105] BASILI, V., TESORIERO, R., COSTA, P., *et al.*, "Building an Experience Base for Software Engineering: A Report on the First CeBASE eWorkshop". In: *Proceedings of Third International Conference on Product Focused Software Process Improvement*, v. 2188, *Lecture Notes in Computer Science*, Springer-Verlag, pp. 110-125, 2001.
- [106] BASILI, V., CALDIERA, G., ROMBACH, H., "Goal Question Metric Paradigm". In: *Encyclopedia of Software Engineering*, v. 1, John Wiley & Sons, pp. 528-532, 1994.

- [107] FELDMANN, R., NICK, M., FREY, M., "Towards Industrial-Strength Measurement Programs for Reuse and Experience Repository Systems". In: *Proceedings of the 2nd Workshop on Learning Software Organizations*, pp. 7-18, Oulu, Finland, Jun. 2000.
- [108] SEAMAN, C., MENDONÇA, M., BASILI, V. *et al.*, "An Experience Management System for a Software Consulting Organization". In: *Proceedings of the 24th Annual Software Engineering Workshop*, Software Engineering Laboratory, Goddard Space Flight Center - NASA, Greenbelt, USA, Dec. 1999.
- [109] TAUTZ, C., WANGENHEIM, C., *REFSENO: A representation formalism for software engineering ontologies*, IESE-Report No. 015.98/E, Fraunhofer Institute for Experimental Software Engineering, Kaiserslautern, Germany, 1998.
- [110] JOHANSSON, C., HALL, P., COQUARD, M., "Talk to Paula and Peter – They are Experienced". In: *Proceedings of the Workshop on Learning Software Organizations*, pp. 69-76, Kaiserslautern, Germany, Jun. 1999.
- [111] CREIGHTON, J. L., ADAMS, J. W., *Cyber Meeting: How to Link People and Technology in your Organization*, New York, USA, AMACOM, 1998. Ref: LUCCA *et al.* (2000).
- [112] KING, W., MARKS, P., MCCOY, S., "The Most Important Issues in Knowledge Management", *Communications of the ACM*, v. 45, n. 9, pp. 93-97, Sep. 2002.
- [113] HENNINGER, S., "Using Software Process to Support Learning Software Organizations". In: *Proceedings of the Workshop on Learning Software Organizations*, pp. 99-114, Kaiserslautern, Germany, Jun. 1999.
- [114] HENNINGER, S., SCHLABACH, J., "A Tool for Managing Software Development Knowledge", In: *Proceedings of Third International Conference on Product Focused Software Process Improvement*, v. 2188, *Lecture Notes in Computer Science*, Springer-Verlag, pp. 182-195, 2001.
- [115] GALOTTA, C., ZANETTI, D., SOUZA, R., *et al.* "CORE-KM: um Ambiente Customizável para Gerência de Conhecimento". In: *Anais do II Simpósio Brasileiro de Qualidade de Software*, Fortaleza, Brasil, Set. 2003.
- [116] MOURA, L., *Taxonomia de Ambientes de Desenvolvimento de Software*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 1992.
- [117] HARRISON, W., OSSHER, H., TARR, P., "Software Engineering Tools and Environments: A Roadmap". In: *Proceedings of the Conference on the Future of Software engineering - International Conference on Software Engineering*, pp. 261-277, Limerick, Ireland, 2000.

- [118] PENEDO, M., *Towards understanding Software Engineering Environment*, Technical Report IMPSEE-TRW-93-003, 1993.
- [119] SLISKI, T., BILLMERS, M. P., CLARKE, L. A., *et al.*, "An Architecture for Flexible, Evolvable Process-Driven User-Guidance Environments". In: *Proceedings of the 8th European Software Engineering Conference – 9th ACM SIGSOFT Symposium on Foundations of Software Engineering*, pp. 33-43, Vienna, Austria, Sep. 2001.
- [120] OSTERWEIL, L. J., "A Process-Object Centered View of Software Environment Architecture". In: *Advanced Programming Environments*, v. 244, *Lecture Notes in Computer Science*, Springer-Verlag, pp. 156-174, 1986. Ref: SLISKI *et al.* (2001).
- [121] BEAUDOUIN-LAFON, M., "User Interface Support for the Integration of Software Tools: An Iconic Model of Interaction". In: *Proceedings of the ACM SIGSOFT/SIGPLAN Software Engineering Symposium on Practical Software Development Environments*, pp. 143-152, Boston, USA, Nov.1988. Ref: SLISKI *et al.* (2001).
- [122] WASSERMAN, A. I., "Tool Integration in Software Engineering Environments" . In: *Proceedings of the International Workshop on Software Engineering Environments*, pp. 137-149, Chinon, France, 1989. Ref: SLISKI *et al.* (2001).
- [123] FALBO, R., *Integração de Conhecimento em um Ambiente de Desenvolvimento de Software*, Tese de D. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 1998.
- [124] KOBIALKA, H., *Implementing Support for Software Processes in a Process-centered Software Engineering Environment*, Technical Report Number 15, GMD Research Series, Sankt Augustin, Germany, 1998.
- [125] ARBAOUI, S., DERNIAME, J., OQUENDO, F., *et al.*, "A Comparative Review of Process-Centered Software Engineering Environments", *Annals of Software Engineering*, v. 14, n. 1-4, pp. 311-340, Dec. 2002
- [126] VILLELA, K., WERNER, C. M., *Framework para Comparação de Ambientes de Desenvolvimento de Software Orientados a Domínio*, Publicações Técnicas 1/98, COPPE/UFRJ, Rio de Janeiro, Brasil, 1998.
- [127] FUGGETTA, A., "Software Process: a roadmap", In: *Proceedings of the Conference on the Future of Software Engineering - International Conference on Software Engineering*, pp. 25-34, Limerick, Ireland, 2000.
- [128] GARG, P., JAZAYERI, M., *Process-centered Software Engineering Environments*, IEEE Computer Society Press, 1996.

- [129] DOWSON, M., "Consistency Maintenance in Process Sensitive Environments". In *Proceedings of the Process Sensitive SEE Architectures Workshop*, Boulder, USA, Sep. 1992. Ref: ARBAOUI *et al.* (2002).
- [130] ARBAOUI, S., OQUENDO, F., "Managing Inconsistencies between Process Enactment and Process Performance States". In: *Proceedings of the 8th International Software Process Workshop*, pp. 24-27, Wadern, Germany, Mar. 1993. Ref: ARBAOUI *et al.* (2002).
- [131] MAURER, F., DELLEN, B., "Process Support for Virtual Software Organizations", In: *Proceedings of the Workshop on Learning Software Organizations*, pp. 87-96, Kaiserslautern, Germany, Jun. 1999.
- [132] WfMC Work Group 1, *Interface 1: Process Definition Interchange – Process Model*, TC00-1016-P, version 1.1, Workflow Management Coalition, 1999.
- [133] CUGOLA, G., GHEZZI, C., "Design and Implementation of PROSYT: a Distributed Process Support System". In: *Proceedings of the 8th Workshop on Enabling Technologies on Infrastructure for Collaborative Enterprises*, pp. 32-39, Palo Alto, Jun. 1999.
- [134] REIS, C., REIS, R., SCHLEBBE, H., *et al.*, "A Policy-based Resource Instantiation Mechanism to Automate Software Process Management". In: *Proceedings of 14th International Conference on Software Engineering & Knowledge Engineering*, pp. 795-802, Ischia, Italy, Jul. 2002.
- [135] MAURER, F., HOLZ, H., "Integrating Process Support and Knowledge Management for Virtual Software Development Teams", *Annals of Software Engineering*, v. 14, n. 1-4, pp. 145-168, Dec. 2002.
- [136] OLIVEIRA, K., ZLOT, F., ROCHA, A. R., *et al.*, "Domain-oriented software development environment", *Journal of Systems and Software* (Article in Press), 2004. In: <http://authors.elsevier.com/sd/article/S0164121203002334>.
- [137] ZLOT, F., *Conhecimento de Tarefa em Ambientes de Desenvolvimento de Software Orientados a Domínio*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2002.
- [138] VASCONCELOS, F., *Reutilização de Processos de Desenvolvimento de Software baseada em Padrões*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 1997.
- [139] ARAÚJO, M. A. P., *Automatização do Processo de Desenvolvimento de Software nos Ambientes Instanciados pela Estação TABA*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 1998.

- [140] ISO/IEC, *ISO/IEC 12207: Information Technology – Software Life Cycle Process*, International Organization for Standardization and International Electrotechnical Commission, Geneva, Switzerland, 1995.
- [141] EMAM, K. E., DROUIN, J., MELO W., *SPICE – The Theory and Practice of Software Process Improvement and Capability Determination*, IEEE Computer Society Press, 1998.
- [142] MAIDANTCHIK, C., *Modelo de Processo de Gerência de Software para Equipes Geograficamente Dispersas*, Tese de D. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 1999.
- [143] ROCHA, A. R., MAIDANTCHIK, C., OLIVEIRA *et al.*, *Experience in Defining, Using and Improving Software Process*, Publicações Técnicas ES 507/99, COPPE/UFRJ, Rio de Janeiro, Brasil, 1999.
- [144] MACHADO, L. F., 2000, *Modelo para Definição, Especialização e Instanciação de Processos de Software na Estação TABA*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2000.
- [145] PAULK, M. C., WEBER, C. V., CURTIS, B. (eds), *The Capability Maturity Model: Guidelines for Improving the Software Process*, Carnegie Mellon University, Software Engineering Institute, Addison-Wesley Longman, 1995.
- [146] CERQUEIRA, A., OLIVEIRA, K., ROCHA, A. R., “Apoio Automatizado para a Definição de Requisitos de Qualidade de Software utilizando Teoria Fuzzy”. In: *Anais do Workshop de Qualidade - XIII Simpósio Brasileiro de Engenharia de Software*, Florianópolis, Brasil, Out., 1999.
- [147] BELCHIOR, A. D., *Um Modelo Fuzzy para Avaliação da Qualidade de Software*, Tese de D. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 1997.
- [148] GALOTTA, C., *Netuno: Um Ambiente de Desenvolvimento de Software Orientado ao Domínio de Acústica Submarinha*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2000.
- [149] GOMES, A. G., *Avaliação de Processos de Software Baseada em Medições*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2001.
- [150] WERNECK, V. M., *Ambiente para Desenvolvimento de Software Baseado em Conhecimento*, Dissertação de D. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 1995.
- [151] WERNER, C. M., TRAVASSOS, G. H., ROCHA, A. R., *et al.*, “Memphis: A Reuse Based O. O. Software Development Environment”. In: *Proceedings of TOOLS*, Beijing, China, Sep. 1997.

- [152] FOURO, A. M., *Apoio à Construção de Base de Dados de Pesquisa em Ambientes de Desenvolvimento de Software Orientados a Domínio*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2002.
- [153] BERTOLLO, G., RUY, F., MIAN, P., *et al.*, "ODE – Um Ambiente de Desenvolvimento de Software Baseado em Ontologias". In: *Anais da IX Sessão de Ferramentas - XVI Simpósio Brasileiro de Engenharia de Software*, pp. 438-443, Gramado, Brasil, Out. 2002.
- [154] BERTOLLO, G., FALBO, R., "Apoio Automatizado à Definição de Processos em Níveis". In: *Anais do II Simpósio Brasileiro de Qualidade de Software*, pp. 77-91, Fortaleza, Brasil, Set. 2003.
- [155] NATALI, A. C., *Uma Infra-Estrutura para Gerência de Conhecimento em um Ambiente de Desenvolvimento de Software*, Dissertação de M. Sc., UFES, Vitória, Brasil, 2003.
- [156] MIAN, P., FALBO, R., "Supporting Ontology Development with ODEd". In: *2ª Jornadas Ibero-Americanas de Ingeniería del Software e Ingeniería del Conocimiento*, Salvador, Brasil, Out. 2002.
- [157] AOYAMA, M., "Web-Based Agile Software Development", *IEEE Software*, v. 15, n. 6, pp. 56-65, Nov/Dec. 1998.
- [158] KOLANO, P., DANG, Z., KEMMERER, R., "The design and analysis of real-time systems using the ASTRAL software development environment", *Annals of Software Engineering*, v.7, n. 1-4, pp. 177-210, 1999.
- [159] DOSSICK, S., KAISER, G., "CHIME: a metadata-based distributed software development environment". In: *Proceedings of the 7th European Software Engineering Conference – 7th ACM SIGSOFT Symposium on Foundations of Software Engineering*, pp. 464-475, Toulouse, France, Sep. 1999.
- [160] GOMAA, H., FARRUKH, G. A., "Methods and tools for the automated configuration of distributed applications from reusable software architectures and components", *IEE Proceedings - Software*, v. 146, n. 6, pp. 277-285, Dec. 1999.
- [161] BRAGA, R., WERNER, C., MATTOSO, M., "Odyssey: A Reuse Environment based on Domain Models". In: *Proceedings of 2nd IEEE Symposium on Application-Specific Systems and Software Engineering Technology*, pp. 49-57, Richardson, USA, Mar. 1999.
- [162] ROSSI, S., MARTTIIN, P., "Adoption of Integrated Process and Product Support for Software Engineering in SP Jyväskylä". In: *International Conference on software Methods and Tools*, pp. 97-105, Wollongong, Australia, Nov. 2000.

- [163] WANG, R., WANG., L, GENG, C, *et al.*, "The Design of VPP Software Development Environment". In: *IEEE Instrumentation and Measurement Technology Conference*, pp. 403-408, Anchorage, USA, May. 2002.
- [164] ABREU, M., SCHLEBBE, H., REIS, C., *et al.*, "APSEE: Uma Abordagem Integrada para Automação da Gerência do Processo de Software". In: *Anais da X Sessão de Ferramentas - XVII Simpósio Brasileiro de Engenharia de Software*, pp. 91-96, Manaus, Brasil, Out. 2003.
- [165] VILLELA, K., ZLOT, F., SANTOS, G, *et al.*, Knowledge Management in Software Development Environments. In: *14th International Conference - Software & Systems Engineering and their Applications*, Paris, France, 2001.
- [166] VILLELA, K., SANTOS, G., TRAVASSOS, G. H., *et al.*, "Gestão de Conhecimento em Ambientes de Desenvolvimento de Software". In: *2^a Jornadas Ibero-Americanas de Ingeniería del Software e Ingeniería del Conocimiento*, Salvador, Brasil, 2002.
- [167] TREACY, M., WIERSEMA, F., *The Discipline of Market Leaders: Choose Your Customers, Narrow Your Focus, Dominate Your Market*, Reading, USA, Addison-Wesley, 1995. Ref: KOUWENHOVEN (1998).
- [168] KITCHENHAM, B., TRAVASSOS, G. H., MAYRHAUSER, A., *et al.*, "Towards an Ontology of Software Maintenance", *Journal of Software Maintenance: Research and Practice*, v. 11, n. 6, pp. 365-389, Nov/Dec. 1999.
- [169] DIAS, M., *Organização do Conhecimento Utilizado na Manutenção de Software*, Dissertação de M. Sc., Universidade Católica de Brasília, Brasília, Brasil, 2003.
- [170] DUARTE, K. C., *Um Framework de Reuso no Domínio da Qualidade de Software*, Dissertação de M. Sc., Universidade Federal do Espírito Santo, Vitória, 2001.
- [171] OMG, *Software Process Engineering – Metamodel Specification*, formal/02-11-14, version 1.0, Object Management Group, 2002.
- [172] RATIONAL, "Business Modeling with UML and Rational Suite AnalystStudio", *A Rational White Paper*, 2001.
- [173] HOLLINGSWORTH, D., *The Workflow Reference Model*, TC00-1003, version 1.1, Workflow Management Coalition, 1995.
- [174] CHANG, Y., CHEN, S., CHEN, C., *et al.*, "Workflow Process Definition and Their Applications in e-Commerce". In: *2000 International Symposium on Multimedia Software Engineering*, pp. 193-200, Taipei, Taiwan, Nov. 2000.

- [175] EMAM, K., "Software Engineering Process". In: Pierre Bourque e Robert Dupuis (eds.), *Guide to the Software Engineering Body of Knowledge*, trial version 0.95, chapter 9, Software Engineering Coordinating Committee, 2001.
- [176] ISO/IEC, *ISO/IEC TR 15504: Information Technology – Software Process Assessment*, Parts 1-9, International Organization for Standardization and the International Electrotechnical Commission, Geneva, Switzerland, 1998-2003.
- [177] SEI, *Capability Maturity Model Integration (CMMI) Version 1.1 - Staged Representation*, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, 2002.
- [178] NOGUEIRA, M., ROCHA, A. R., "Práticas Relevantes em Engenharia de Software: uma avaliação de especialistas". In: *Anais do II Simpósio Brasileiro de Qualidade de Software*, Fortaleza, Brasil, Set. 2003.
- [179] KRUCHTEN, P., *The Rational Unified Process: An Introduction*, Object Technology Series, Addison-Wesley, 2000.
- [180] FOX, M. S., CHIONGLO, J. F., FADEL, F. G., "A Common-Sense Model of the Enterprise". In: *Proceedings of the 2nd Industrial Engineering Research Conference*, pp. 425-429, Norcross, USA, 1993.
- [181] GRUNINGER, M., FOX, M. S., "An Activity Ontology for Enterprise Modelling". In: *Proceedings of the Third Workshop on Enabling Technologies – Infrastructures for Collaborative Enterprises*, West Virginia University, USA, 1994.
- [182] FADEL, F. G., FOX, M. S., GRUNINGER, M., "A Generic Enterprise Resource Ontology". In: *Proceedings of the Third Workshop on Enabling Technologies – Infrastructures for Collaborative Enterprises*, West Virginia University, USA, 1994.
- [183] FOX, M. S., BARBUCEANU, M., GRUNINGER, M., "An Organisation Ontology for Enterprise Modelling: Preliminary Concepts for Linking Structure and Behaviour", *Computers in Industry*, v. 29, pp. 123-134, 1996.
- [184] FOX, M. S., GRUNINGER, M., "Enterprise Modelling", *AI Magazine*, v. 9, n. 3, pp. 109-121, Fall 1998.
- [185] USCHOLD, M., KING, M., MORALEE, S., *et al.*, "The Enterprise Ontology", *The Knowledge Engineering Review*, v.13, n. 1, Special Issue on Putting Ontologies to Use (eds. Mike Uschold e Austin Tate), 1998.
- [186] COTA, R., *Modelagem Computacional para Gestão Empresarial*, Dissertação de M. Sc., Universidade Federal do Espírito Santo, Vitória, Brasil, 2002.

- [187] CHIAVENATO, I., "Decorrências da Abordagem Neoclássica: Tipos de Organização", *Introdução à Teoria Geral da Administração*, 5ª edição, capítulo 8, Editora Campus, 1998a
- [188] CHIAVENATO, I., "Decorrências da Abordagem Neoclássica: Departamentalização", *Introdução à Teoria Geral da Administração*, 5ª edição, capítulo 9, Editora Campus, 1998b.
- [189] CHIAVENATO, I., "Administração por Objetivos (APO)", *Introdução à Teoria Geral da Administração*, 5ª edição, capítulo 10, Editora Campus, 1998c.
- [190] MEGGINSON, L., MOSLEY, D., PIETRI, P., "Estabelecimento de Objetivos e Metas Organizacionais", *Administração: Conceitos e Aplicações*, capítulo 6, Editora Harbra, 1986a.
- [191] MIAN, P. G., ODEd: *Uma Ferramenta de Apoio ao Desenvolvimento de Ontologias em um Ambiente de Desenvolvimento de Software*, Dissertação de M. Sc., Universidade Federal do Espírito Santo, Vitória, Brasil, 2003.
- [192] HOLANDA, A. B., *Novo Aurélio Século XXI - O Dicionário da Língua Portuguesa*, Nova Fronteira, 1999.
- [193] MICHAELIS, *Michaelis Moderno Dicionário da Língua Portuguesa*, Editora Melhoramentos, 1998.
- [194] WOHLIN, C., RUNESON, P., HÖST, M., *et al.*, *Experimentation in Software Engineering: An Introduction*, *The Kluwer International Series in Software Engineering*, Norwell, USA, Kluwer Academic Publishers, 2000.
- [195] PFLEEGER, S., KITCHENHAM, B., "Principles of survey research – Part 1: Turning Lemons into Lemonade", *ACM SIGSOFT Software Engineering Notes*, v. 26, n. 6, pp. 16-18, Nov. 2001.
- [196] KITCHENHAM, B., PFLEEGER, S., "Principles of survey research – Part 2: Designing a Survey", *ACM SIGSOFT Software Engineering Notes*, v. 27, n. 1, pp. 18-20, Jan. 2002.
- [197] KITCHENHAM, B., PFLEEGER, S., "Principles of survey research – Part 3: Constructing a Survey Instrument", *ACM SIGSOFT Software Engineering Notes*, v. 27, n. 2, pp. 20-24, Mar. 2002.
- [198] KITCHENHAM, B., PFLEEGER, S., "Principles of survey research – Part 4: Questionnaire Evaluation", *ACM SIGSOFT Software Engineering Notes*, v. 27, n. 3, pp. 20-23, May 2002.
- [199] KITCHENHAM, B., PFLEEGER, S., "Principles of survey research – Part 5: Populations and Samples", *ACM SIGSOFT Software Engineering Notes*, v. 27, n. 5, pp. 17-20, Sep. 2002.

- [200] KITCHENHAM, B., PFLEEGER, S., "Principles of survey research – Part 6: Data Analysis", *ACM SIGSOFT Software Engineering Notes*, v. 28, n. 2, pp. 24-27, Mar. 2003.
- [201] LETHBRIDGE, T. C., "What Knowledge Is Important to a Software Professional", *IEEE Computer*, v. 33, n. 5, pp. 44-50, May 2000.
- [202] SEPIN/MCT, *Qualidade e Produtividade no Setor de Software Brasileiro 1999*, Secretaria de Política de Informática/Ministério de Ciência e Tecnologia, Brasília, 2000.
- [203] SEPIN/MCT, *Qualidade e Produtividade no Setor de Software Brasileiro 2001*, Secretaria de Política de Informática/Ministério de Ciência e Tecnologia, Brasília, 2002.
- [204] SEPIN/MCT, *Levantamento do Universo de Associadas SOFTEX*, Pesquisa Censo SW – Agosto de 2001, Secretaria de Política de Informática/Ministério de Ciência e Tecnologia, Brasília, 2001.
- [205] PRESSMAN, R., *Software Engineering – A Practitioner's Approach*, 4th edition, USA, McGraw-Hill, 1997.
- [206] PFLEEGER, S., *Software Engineering – Theory and Practice*, New Jersey, USA, Prentice-Hall, 1998.
- [207] JONES C., "Software Benchmarks and Baselines", *Software Assessments, Benchmarks, and Best Practices*, chapter 3, Addison-Wesley, 2000.
- [208] JONES C., "Introduction", *Software Assessments, Benchmarks, and Best Practices*, chapter 1, Addison-Wesley, 2000.
- [209] ITAB, *Project Size and Complexity Calculation Form*, Information Technology Advisory Board, Office of Information Technology, Missouri State Government. In: <http://oit.mo.gov/itab/motions/Project Size and Complexity Calculation Form.xls>, Feb. 2004.
- [210] FATEEV, M., *O Meta-Gerador de Ferramentas da Estação TABA*, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 1998.
- [211] LEVIN, J., *Estatística Aplicada às Ciências Humanas*, São Paulo, Harper & Row, 1987.
- [212] ISO/IEC, *ISO/IEC 14598-5 Information Technology – Software Product Evaluation - Parte 4: Process for Acquirers*, 1st edition, International Organization for Standardization and the International Electrotechnical Commission, 1998.

ANEXO I

Instrumentação da Pesquisa: Questionários



A (nome do responsável), (cargo na empresa)

Desenvolvedores de software lidam com diferentes tipos de conhecimento ao longo do processo de desenvolvimento de software. Em geral, conhecimento importante neste contexto encontra-se disperso nas mentes de várias pessoas e/ou em documentos armazenados em vários lugares e mídias. Conseqüentemente, desenvolvedores de software, muitas vezes, falham em atender rápida e adequadamente às solicitações, porque o conhecimento de que necessitam foi perdido pela organização ou está em lugar desconhecido ou inacessível.

Para melhorar a maneira como as organizações desenvolvem e mantêm software, é fundamental melhorar a maneira como elas administram o conhecimento necessário para a realização destas atividades.

Com o intuito de identificar quais tipos de conhecimento são considerados mais importantes no contexto de ambientes para desenvolvimento de software, estamos aplicando um questionário em empresas de Salvador. A (nome da empresa) foi uma das empresas selecionadas.

Desta forma, solicitamos algumas informações básicas sobre a empresa e a participação de seus desenvolvedores de software na nossa pesquisa. A identificação da empresa e dos desenvolvedores não serão divulgadas. Comprometemo-nos a fornecer para a empresa o resultado geral da pesquisa e o resultado parcial obtido na mesma.

1. Perfil da Empresa

1.01 Área de atuação da empresa: _____

1.02 Estado em que a matriz está localizada: _____

1.03 Atividade(s) de desenvolvimento de software da empresa:

01. Desenvolvimento de software para uso próprio
 02. Desenvolvimento de software para uso próprio com contratação de terceiros
 03. Desenvolvimento de pacote de software para comercialização (*packaged software*)
 04. Desenvolvimento de software sob encomenda para terceiros (*custom software*)

1.04 Como classificaria a rotatividade de pessoal na empresa?

01. Baixa 02. Baixa a Média 03. Média 04. Média a Alta 05. Alta

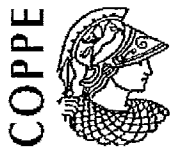
1.05 Como classificaria a rotatividade de pessoal na empresa (área de software)?

01. Baixa 02. Baixa a Média 03. Média 04. Média a Alta 05. Alta

Desde já, agradecemos a colaboração de sua empresa.

Karina Villela de Carvalho Lima

Grupo de Engenharia de Software da COPPE/UFRJ.



Programa de Engenharia de Sistemas e Computação
Coordenação dos Programas de Pós-Graduação de Engenharia
Universidade Federal do Rio de Janeiro
Cidade Universitária, Centro de Tecnologia - Bloco H, Sala 319
Caixa Postal 68511, Rio de Janeiro, RJ, CEP 21945-970
Tel: (021) 562-8698
Fax: (021) 290-6626
www:http://www.cos.ufrj.br

Pesquisa em Ambiente para Desenvolvimento de Software

Desenvolvedores de software lidam com diferentes tipos de conhecimento ao longo do processo de desenvolvimento de software. Em geral, conhecimento importante neste contexto encontra-se disperso nas mentes de várias pessoas e/ou em documentos armazenados em vários lugares e mídias. Conseqüentemente, desenvolvedores de software, muitas vezes, falham em atender rápida e adequadamente às solicitações, porque o conhecimento que necessitam foi perdido pela organização ou está em lugar desconhecido ou inacessível.

Quando o conhecimento necessário para uma tarefa não está disponível, o desenvolvedor de software tem que partir do zero para a solução do problema, o que tem implicações na qualidade da solução, no tempo e no custo necessários para obtê-la. Além disso, erros cometidos anteriormente podem ser novamente cometidos.

Para melhorar a maneira como as organizações desenvolvem e mantêm software, é fundamental melhorar a maneira como elas administram o conhecimento requerido para a realização destas atividades. Nossa proposta é identificar e prover esse conhecimento como parte de um ambiente para desenvolvimento de software.

Desenvolvedores de software alocados a seu projeto irão participar desta pesquisa. Desta forma, solicitamos a sua participação como gerente de projeto, fornecendo algumas informações básicas sobre o mesmo. A sua identificação e a identificação de sua empresa não serão divulgadas.

Contamos com o seu conhecimento e experiência. A qualidade da nossa pesquisa será tão boa quanto a qualidade de suas respostas. Desde já, agradecemos a sua colaboração.

Grupo de Engenharia de Software da COPPE/UFRJ.

ID: _____

Autorizo o uso das informações por mim fornecidas no contexto desta pesquisa desde que a minha identificação e a da minha empresa não sejam divulgadas.

Nome: _____

Endereço eletrônico: _____

Nome da empresa: _____

2. Perfil do Projeto

2.01 Nome do projeto de software: _____

2.02 Tamanho previsto do projeto de software:

01. Duração: _____ 02. Número de pessoas na equipe: _____

2.03 Objetivo(s) do projeto de software:

01. Novo desenvolvimento 02. Adequação ao contexto 03. Reparo de defeitos 04. Adição de funções
 05. Migração de plataforma 06. Melhoria de performance 07. Reengenharia 08. Atualização em massa

2.04 Classe(s) que melhor caracteriza(m) o projeto de software:

01. Aplicação para uso corporativo desenvolvida apenas com recursos internos
 02. Aplicação para uso corporativo desenvolvida com a contratação de terceiros
 03. Aplicação para uso externo via Internet
 04. Aplicação para controle de dispositivos físicos
 05. Aplicação comercial para venda ou *leasing*
 06. Aplicação desenvolvida sob contrato de encomenda para terceiros
 07. Aplicação que obedece a padrões militares

2.05 Este projeto envolve apoio à tomada de decisão gerencial na empresa?

01. Sim 02. Não

2.06 Como você classificaria a rotatividade de pessoal neste projeto de software?

01. Baixa 02. Baixa a Média 03. Média 04. Média a Alta 05. Alta

2.07 Cite alguns fornecedores de produtos e/ou serviços¹ da empresa envolvidos no projeto?
_____2.08 Cite alguns parceiros técnicos² da empresa envolvidos no projeto?

2.09 Cite alguns clientes da empresa envolvidos no projeto? _____

Solicitamos fornecedores, parceiros técnicos e clientes apenas para que os desenvolvedores do projeto tenham uma visão homogênea do projeto, mas não temos interesse nestas informações. Elas não são obrigatórias. Informe-as se considerar que os desenvolvedores podem ter dificuldade em reconhecê-los e, conseqüentemente, atribuir importância equivocada às informações relativas a eles.

¹ **Fornecedor de Produto e/ou Serviço:** organização que tem um contrato ou acordo com a outra organização para fornecimento de produtos e/ou serviços. Estes produtos e serviços podem estar ou não relacionados a área de desenvolvimento de software. Nenhuma transferência de tecnologia é realizada.

² **Parceiro Técnico:** organização que tem um contrato ou acordo com a outra organização para a realização conjunta de atividades, dividindo a responsabilidade e os resultados destas atividades. As atividades podem estar ou não relacionados a área de desenvolvimento de software. A organização que é parceira técnica contribui com o conhecimento que domina, podendo haver transferência de tecnologia.



Programa de Engenharia de Sistemas e Computação
Coordenação dos Programas de Pós Graduação de Engenharia
Universidade Federal do Rio de Janeiro
Cidade Universitária, Centro de Tecnologia - Bloco H, Sala 319
Caixa Postal 68511, Rio de Janeiro, RJ, CEP 21945-970
Tel: (021) 562-8698
Fax: (021) 290-6626
www: <http://www.cos.ufrj.br>

Pesquisa em Ambiente para Desenvolvimento de Software

Desenvolvedores de software lidam com diferentes tipos de conhecimento ao longo do processo de desenvolvimento de software. Em geral, conhecimento importante neste contexto encontra-se disperso nas mentes de várias pessoas e/ou em documentos armazenados em vários lugares e mídias. Conseqüentemente, desenvolvedores de software, muitas vezes, falham em atender rápida e adequadamente às solicitações, porque o conhecimento que necessitam foi perdido pela organização ou está em lugar desconhecido ou inacessível.

Quando o conhecimento necessário para uma tarefa não está disponível, o desenvolvedor de software tem que partir do zero para a solução do problema, o que tem implicações na qualidade da solução, no tempo e no custo necessários para obtê-la. Além disso, erros cometidos anteriormente podem ser novamente cometidos.

Para melhorar a maneira como as organizações desenvolvem e mantêm software, é fundamental melhorar a maneira como elas administram o conhecimento requerido para a realização destas atividades. Nossa proposta é identificar e prover esse conhecimento como parte de um ambiente para desenvolvimento de software.

Para isto, solicitamos sua participação nesta pesquisa. A sua identificação e a identificação de sua empresa não serão divulgadas.

Contamos com o seu conhecimento e experiência. A qualidade da nossa pesquisa será tão boa quanto a qualidade de suas respostas. Desde já, agradecemos a sua colaboração.

Grupo de Engenharia de Software da COPPE/UFRJ.

ID: _____

Autorizo o uso das informações por mim fornecidas no contexto desta pesquisa desde que a minha identificação e a da minha empresa não sejam divulgadas.

Nome: _____

Endereço eletrônico: _____

3. Perfil Profissional**3.01 Grau de escolaridade (especifique a área quando pertinente):**

01. 2º grau 02. Em Graduação 03. Graduação 04. Especialização 05. Mestrado 06. Doutorado
 07. Área: _____

3.02 Tempo de experiência na área de desenvolvimento de software: _____ ano(s)**3.03 Número de empresas em que já atuou na área de desenvolvimento de software, incluindo a empresa atual:**

01. 1 02. 2 03. 3 a 5 04. mais de 5

3.04 Nome da empresa onde trabalha atualmente: _____**3.05 Função que exerce atualmente na área de desenvolvimento de software:**

01. Empresário 02. Diretor 03. Gerente/Coordenador de Área
 04. Gerente de Projeto 05. Analista de Sistema 06. Programador

3.06 Nome do projeto do qual participa atualmente: _____

(se está participando de mais de um projeto, escolha o que acha mais interessante no contexto da pesquisa)

3.07 Atividades do ciclo de desenvolvimento de software de que tem participado no projeto atual:

01. Engenharia de Sistemas¹ 02. Planejamento do Projeto 03. Especificação de Requisitos
 04. Projeto 05. Codificação 06. Teste
 07. Implantação 08. Manutenção

3.08 No projeto de software que você participa atualmente, como classificaria sua interação com as pessoas de outras unidades organizacionais (setores, diretorias etc.)?

01. Inexistente 02. Eventual 03. Freqüente

3.09 No projeto de software que você participa atualmente, como classificaria sua interação com os fornecedores de produtos e/ou serviços² da empresa?

01. Inexistente 02. Eventual 03. Freqüente 04. Não se aplica

3.10 No projeto de software que você participa atualmente, como classificaria sua interação com os parceiros técnicos³ da empresa?

01. Inexistente 02. Eventual 03. Freqüente 04. Não se aplica

3.11 No projeto de software que você participa atualmente, como classificaria sua interação com os clientes da empresa?

01. Inexistente 02. Eventual 03. Freqüente 04. Não se aplica

¹ **Engenharia de Sistema:** atividade de solução de problema em que dados, funções e comportamento do produto desejado são descobertos, analisados e alocados a componentes individuais de engenharia (software, hardware, bases de dados e pessoas). Esta atividade envolve a identificação das necessidades do cliente, estudo de viabilidade, alocação de funções a hardware, software, pessoas e outros elementos do sistema, estabelecimento de restrições de custo e cronograma, criação de uma definição do sistema e de sua arquitetura.

² **Fornecedor de Produto e/ou Serviço:** organização que tem um contrato ou acordo com a outra organização para fornecimento de produtos e/ou serviços. Estes produtos e serviços podem estar ou não relacionados a área de desenvolvimento de software. Nenhuma transferência de tecnologia é realizada.

³ **Parceiro Técnico:** organização que tem um contrato ou acordo com a outra organização para a realização conjunta de atividades, dividindo a responsabilidade e os resultados destas atividades. As atividades podem estar ou não relacionados a área de desenvolvimento de software. A organização que é parceira técnica contribui com o conhecimento que domina, podendo haver transferência de tecnologia.

Para o projeto de software do qual participa atualmente, atribua o grau de importância de cada tipo de conhecimento listado abaixo, independente de estar disponível ou não na empresa. Utilize a seguinte escala:

0 - Sem importância, 1 - Pouco importante, 2 – Importante, 3 - Muito importante, 4 - Imprescindível, NA - Não avaliado

A opção NA deve ser utilizada nos casos em que não se considere em condições de avaliar a importância do tipo de conhecimento em questão.

4. Com relação ao domínio de aplicação	
4.01. Conhecimento sobre o domínio da aplicação em desenvolvimento ou manutenção (por exemplo: cardiologia, seguros, telecomunicação etc.)	0 1 2 3 4 NA
5. Com relação à organização	
5.01. Conhecimento sobre:	
01. A missão ⁴ da organização	0 1 2 3 4 NA
02. As restrições que podem dificultar o cumprimento da missão da organização, e sobre suas deficiências e potenciais	0 1 2 3 4 NA
03. As diretrizes e normas organizacionais	0 1 2 3 4 NA
04. Os objetivos e as metas organizacionais	0 1 2 3 4 NA
05. As áreas da organização consideradas críticas para o alcance dos objetivos organizacionais	0 1 2 3 4 NA
06. Os serviços e/ou produtos oferecidos pela organização, incluindo conhecimento sobre as estratégias de marketing, venda e distribuição	0 1 2 3 4 NA
07. A estrutura organizacional, ou seja, conhecimento sobre a distribuição de autoridade e responsabilidade dentro da organização	0 1 2 3 4 NA
08. A alocação dos profissionais da organização às unidades organizacionais	0 1 2 3 4 NA
09. A distribuição de conhecimento e habilidades entre os profissionais da organização	0 1 2 3 4 NA
10. As relações informais que existem entre os profissionais da organização e que podem afetar a distribuição de autoridade e responsabilidade	0 1 2 3 4 NA
11. Os processos organizacionais	0 1 2 3 4 NA
12. Os contratos e acordos firmados entre a organização e seus fornecedores, parceiros técnicos e clientes	0 1 2 3 4 NA
6. Com relação à experiência em desenvolvimento e manutenção de software acumulada pela organização	
6.01. Conhecimento sobre:	
01. Os tipos de software desenvolvidos na organização	0 1 2 3 4 NA
02. O processo de software ⁵ da organização	0 1 2 3 4 NA
03. A avaliação do processo de software da organização, incluindo as áreas do processo a serem melhoradas	0 1 2 3 4 NA
6.02. Modelos de documentos utilizados na organização, incluindo exemplos reais de utilização destes documentos	0 1 2 3 4 NA
6.03. Itens de software ⁶ já produzidos na organização e que podem ser úteis em desenvolvimentos futuros	0 1 2 3 4 NA
6.04. Melhores práticas ⁷ da organização para:	
01. Modelagem de processo	0 1 2 3 4 NA

⁴ **Missão da Organização:** objetivo ou motivo central da existência da organização, definido em termos de satisfazer alguma necessidade do ambiente externo.

⁵ **Processo de Software:** conjunto de atividades, métodos, práticas e transformações que as pessoas empregam para desenvolver, manter, operar, adquirir e/ou fornecer software.

⁶ **Item de Software:** entidade gerada com alguma finalidade durante um processo de software. Itens de software devem ser unicamente identificados, de forma a facilitar a sua reutilização em um novo contexto. Exemplos de itens de software são: plano de projeto, arquitetura do sistema, código fonte e casos de testes.

⁷ **Melhores Práticas:** práticas (métodos, técnicas e ferramentas) utilizadas em um número significativo de projetos de software da organização, as quais foram consideradas benéficas por aqueles que as utilizaram em todos ou na maioria (90%) dos projetos de software.

6.04. Melhores práticas da organização para:						
02. Gerência de projeto	0	1	2	3	4	NA
03. Gerência de risco	0	1	2	3	4	NA
04. Gerência de qualidade	0	1	2	3	4	NA
05. Gerência de configuração	0	1	2	3	4	NA
06. Documentação de software	0	1	2	3	4	NA
07. Engenharia de sistemas	0	1	2	3	4	NA
08. Análise e especificação de requisitos de software	0	1	2	3	4	NA
09. Projeto de software	0	1	2	3	4	NA
10. Codificação	0	1	2	3	4	NA
11. Testes	0	1	2	3	4	NA
12. Operação de sistema (hardware, software etc.) e suporte a usuários	0	1	2	3	4	NA
13. Manutenção de software	0	1	2	3	4	NA
14. Engenharia reversa ⁸ e reengenharia ⁹ de software	0	1	2	3	4	NA
15. Reuso de itens de software	0	1	2	3	4	NA
16. Avaliação e melhoria de processo de software	0	1	2	3	4	NA
17. Treinamento	0	1	2	3	4	NA
18. O processo de aquisição de produto de software	0	1	2	3	4	NA
19. O processo de fornecimento de produto de software	0	1	2	3	4	NA
20. O processo de solução de problemas ¹⁰	0	1	2	3	4	NA
6.05. Lições aprendidas ¹¹ pela organização sobre:						
01. Modelagem de processo	0	1	2	3	4	NA
02. Gerência de projeto	0	1	2	3	4	NA
03. Gerência de risco	0	1	2	3	4	NA
04. Gerência de qualidade	0	1	2	3	4	NA
05. Gerência de configuração	0	1	2	3	4	NA
06. Documentação de software	0	1	2	3	4	NA
07. Engenharia de sistemas	0	1	2	3	4	NA

⁸ **Engenharia Reversa de Software:** processo de analisar um sistema para identificar os seus componentes e a relação entre eles e, assim, obter outras representações do sistema no mesmo nível de abstração ou em um nível de abstração mais alto.

⁹ **Reengenharia de Software:** processo de analisar um sistema e reconstituí-lo numa nova forma com a intenção de melhorar a sua qualidade.

¹⁰ **Processo de Solução de Problemas:** processo para analisar e resolver os problemas, de qualquer natureza ou fonte, que são descobertos durante o desenvolvimento, operação, manutenção ou outros processos.

¹¹ **Lições Aprendidas:** lições aprendidas pela organização em função dos resultados (positivos ou negativos) obtidos na solução de problemas surgidos em projetos de software. Lições aprendidas podem retratar experiências de sucesso, em que os problemas foram resolvidos de forma positiva, e experiências de insucesso, exemplos de soluções que deram errado.

6.05. Lições aprendidas pela organização sobre:	
08. Análise e especificação de requisitos de software	0 1 2 3 4 NA
09. Projeto de software	0 1 2 3 4 NA
10. Codificação	0 1 2 3 4 NA
11. Testes	0 1 2 3 4 NA
12. Operação de sistema (hardware, software etc.) e suporte a usuários	0 1 2 3 4 NA
13. Manutenção de software	0 1 2 3 4 NA
14. Engenharia reversa e reengenharia de software	0 1 2 3 4 NA
15. Reuso de itens de software	0 1 2 3 4 NA
16. Avaliação e melhoria de processo de software	0 1 2 3 4 NA
17. Treinamento	0 1 2 3 4 NA
18. O processo de aquisição de produto de software	0 1 2 3 4 NA
19. O processo de fornecimento de produto de software	0 1 2 3 4 NA
20. O processo de solução de problemas	0 1 2 3 4 NA
6.06. Dados históricos da organização relacionados a:	
01. Gerência de projeto (por exemplo: estimativas e dados reais de produtividade, custo e tempo)	0 1 2 3 4 NA
02. Gerência de risco (por exemplo: probabilidade atribuída e estratégia adotada para riscos em projetos anteriores)	0 1 2 3 4 NA
03. Gerência de qualidade (por exemplo: custo para correção de defeitos, tempo para realização de mudanças)	0 1 2 3 4 NA
04. Treinamento (por exemplo: horas de treinamento, investimento em treinamento, avaliação de desempenho)	0 1 2 3 4 NA
05. Métricas de produto (por exemplo: velocidade de execução, defeitos reportados, complexidade, confiabilidade)	0 1 2 3 4 NA
06. Métricas de processo (por exemplo: número de defeitos, esforço despendido e conformidade com cronograma)	0 1 2 3 4 NA
6.07. Respostas para as perguntas mais freqüentes entre os desenvolvedores de software da organização	
	0 1 2 3 4 NA

7. Com relação aos fornecedores de produtos e/ou serviços da organização	
7.01. Conhecimento sobre:	
01. A missão das empresas fornecedoras de produtos e/ou serviços	0 1 2 3 4 NA
02. As restrições que podem dificultar o cumprimento da missão destas empresas, e sobre suas deficiências e potenciais	0 1 2 3 4 NA
03. As diretrizes e normas organizacionais das empresas fornecedoras de produtos e/ou serviços	0 1 2 3 4 NA
04. Os produtos e/ou serviços oferecidos por estas empresas, incluindo conhecimento sobre as estratégias de marketing, venda e distribuição	0 1 2 3 4 NA
05. A estrutura organizacional das empresas fornecedoras de produtos e/ou serviços, ou seja, sobre a distribuição de autoridade e responsabilidade dentro destas organizações	0 1 2 3 4 NA
06. A alocação de profissionais das empresas fornecedoras a suas respectivas unidades organizacionais	0 1 2 3 4 NA
07. A distribuição de conhecimento e habilidades entre os profissionais das empresas fornecedoras de produtos e/ou serviços	0 1 2 3 4 NA
08. As relações informais que existem entre os profissionais das empresas fornecedoras e que podem afetar a distribuição de autoridade e responsabilidade	0 1 2 3 4 NA
09. Os processos organizacionais das empresas fornecedoras de produtos e/ou serviços	0 1 2 3 4 NA

8. Com relação aos parceiros técnicos da organização

8.01. Conhecimento sobre:	
01. A missão das empresas que são parceiras técnicas	0 1 2 3 4 NA
02. As restrições que podem dificultar o cumprimento da missão destas empresas, e sobre suas deficiências e potenciais	0 1 2 3 4 NA
03. As diretrizes e normas organizacionais das empresas que são parceiras técnicas	0 1 2 3 4 NA
04. Os produtos e/ou serviços oferecidos por estas empresas, incluindo conhecimento sobre as estratégias de marketing, venda e distribuição	0 1 2 3 4 NA
05. A estrutura organizacional das empresas que são parceiras técnicas, ou seja, sobre a distribuição de autoridade e responsabilidade dentro destas organizações	0 1 2 3 4 NA
06. A alocação de profissionais das empresas que são parceiras técnicas a suas respectivas unidades organizacionais	0 1 2 3 4 NA
07. A distribuição de conhecimento e habilidades entre os profissionais das empresas que são parceiras técnicas	0 1 2 3 4 NA
08. As relações informais que existem entre os profissionais das empresas parceiras técnicas e que podem afetar a distribuição de autoridade e responsabilidade	0 1 2 3 4 NA
09. Os processos organizacionais das empresas que são parceiras técnicas	0 1 2 3 4 NA

9. Com relação aos clientes da organização

9.01. Conhecimento sobre:	
01. A missão das empresas clientes	0 1 2 3 4 NA
02. As restrições que podem dificultar o cumprimento da missão das empresas clientes, e sobre suas deficiências e potenciais	0 1 2 3 4 NA
03. As diretrizes e normas organizacionais das empresas clientes	0 1 2 3 4 NA
04. Os produtos e/ou serviços oferecidos pelas empresas clientes, incluindo conhecimento sobre as estratégias de marketing, venda e distribuição	0 1 2 3 4 NA
05. A estrutura organizacional das empresas clientes, ou seja, sobre a distribuição de autoridade e responsabilidade dentro destas organizações	0 1 2 3 4 NA
06. A alocação de profissionais das empresas clientes a suas respectivas unidades organizacionais	0 1 2 3 4 NA
07. A distribuição de conhecimento e habilidades entre os profissionais das empresas clientes	0 1 2 3 4 NA
08. As relações informais que existem entre os profissionais das empresas clientes e que podem afetar a distribuição de autoridade e responsabilidade	0 1 2 3 4 NA
09. Os processos organizacionais das empresas clientes	0 1 2 3 4 NA

10. Com relação à literatura técnica

10.01. Indicação de onde encontrar:	
01. Determinado conhecimento teórico em Engenharia de Software	0 1 2 3 4 NA
02. Referências sobre inovações tecnológicas em Engenharia de Software	0 1 2 3 4 NA
03. Relatos sobre as melhores práticas da indústria de software	0 1 2 3 4 NA
04. Relatos sobre as lições aprendidas pela indústria de software	0 1 2 3 4 NA

Especifique os tipos de conhecimento que não foram listados acima e que você considera importante para o projeto de software do qual participa atualmente. Atribua um grau de importância para cada tipo de conhecimento, utilizando a seguinte escala:

0 - Sem importância, 1 - Pouco importante, 2 - Importante, 3 - Muito importante, 4 - Imprescindível

11. Outros tipos de conhecimento que devem ser considerados					
11.01.					
11.02.					
11.03.					
11.04.					
11.05.					

ANEXO II

**Instrumentação da Pesquisa:
Sistema para Aplicação de
Questionários na Web**

Este anexo exhibe algumas telas e o modelo de classes do sistema desenvolvido para compor a instrumentação da pesquisa descrita no capítulo 6. A depender da forma selecionada pela organização para aplicação do questionário, os preparativos para execução da pesquisa incluíam o cadastramento no sistema de cada pessoa da empresa envolvida com o desenvolvimento e a manutenção de software, pois o acesso ao sistema era controlado. A figura II.1 exhibe a tela em que o desenvolvedor era motivado a participar da pesquisa, sendo também informado do seu objetivo, do tempo médio necessário para preencher o questionário da pesquisa e de como a questão da confiabilidade seria tratada. Após se identificar, o desenvolvedor tinha acesso ao questionário.

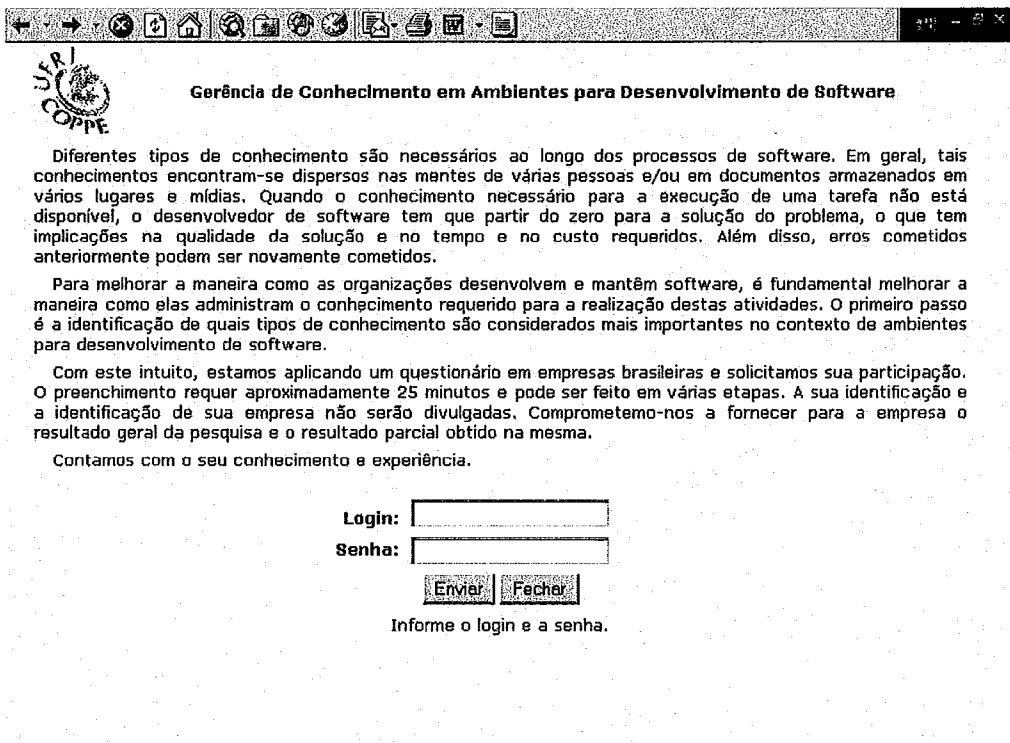


Figura II.1 – Tela de Identificação do Participante

A figura II.2 exhibe a tela que fornecia ao participante as instruções para preenchimento do questionário. Essas instruções indicavam para o participante se ele tinha sido selecionado para caracterização da empresa e/ou de projetos e com relação a qual projeto específico ele iria fornecer a sua opinião sobre a importância dos tipos de conhecimento listados. A figura II.3 mostra a tela para caracterização de um ou mais projetos. A exclamação vermelha no botão *Perfil do Projeto* (figuras II.2 a II.4) indicava que existiam questões na seção que ainda não tinham sido respondidas. A figura II.4 exhibe a tela de uma das seções da parte principal do questionário. Por fim, o modelo de classes do sistema é apresentado na figura II.5.

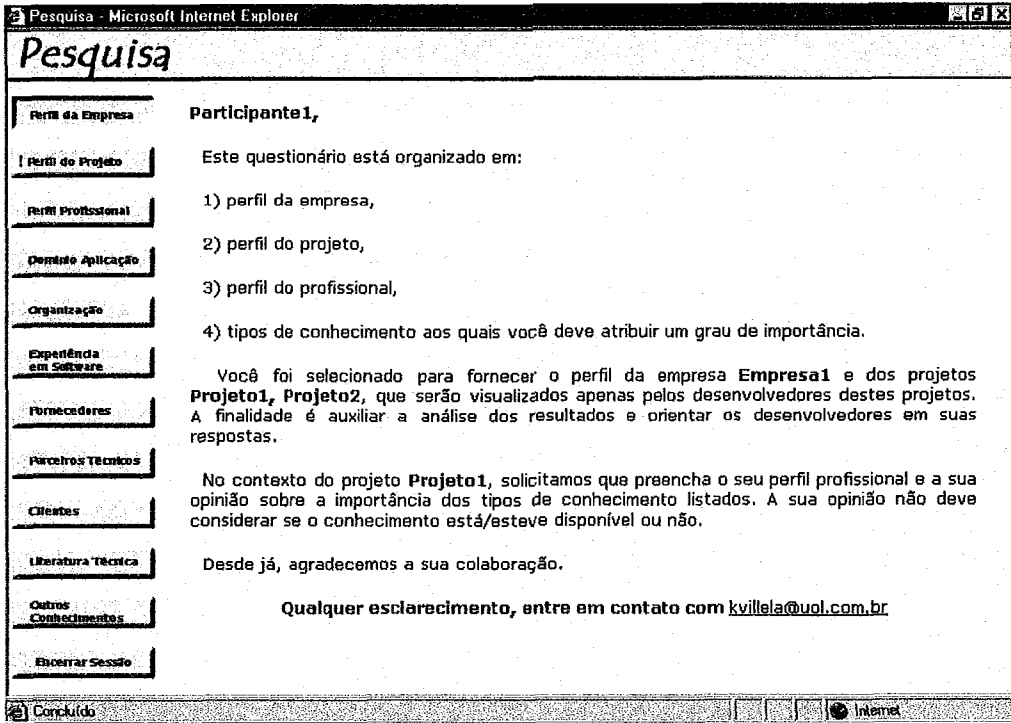


Figura II.2 – Tela Inicial com Instruções de Preenchimento

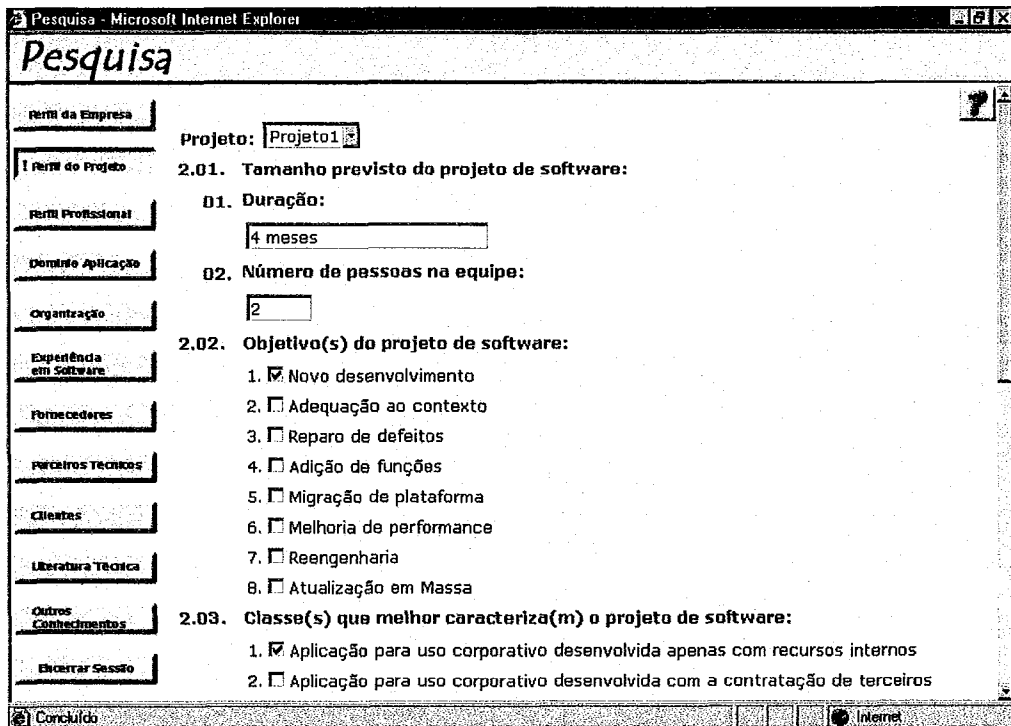


Figura II.3 – Tela para Caracterização do Projeto

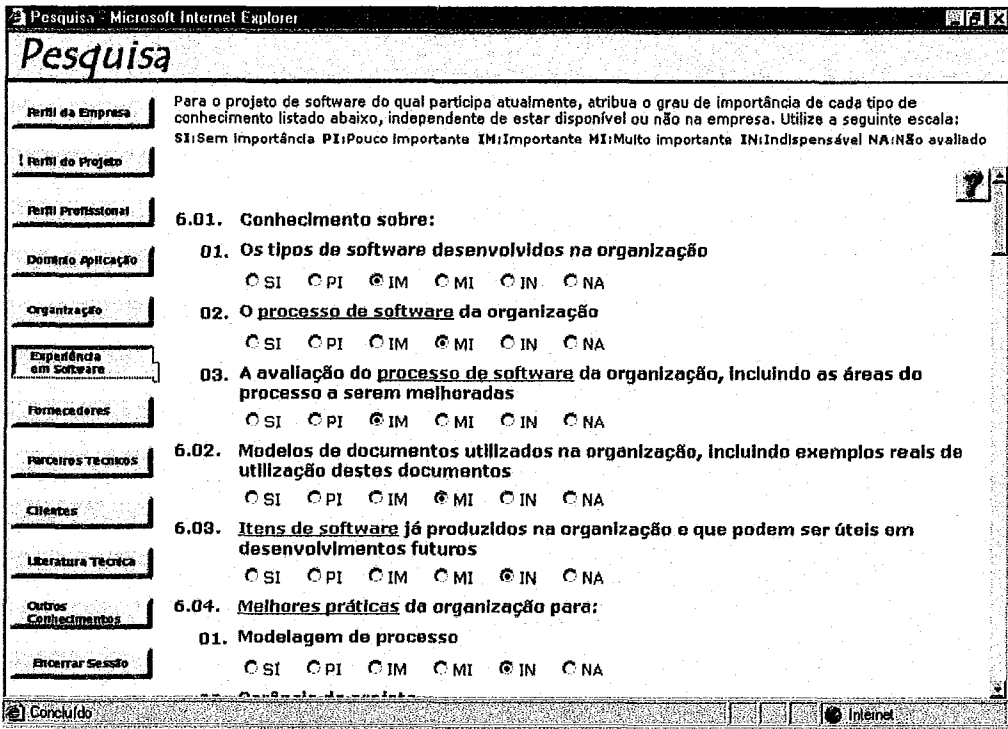


Figura II.4 – Tela referente aos Conhecimentos resultantes da Experiência em Engenharia de Software da Organização

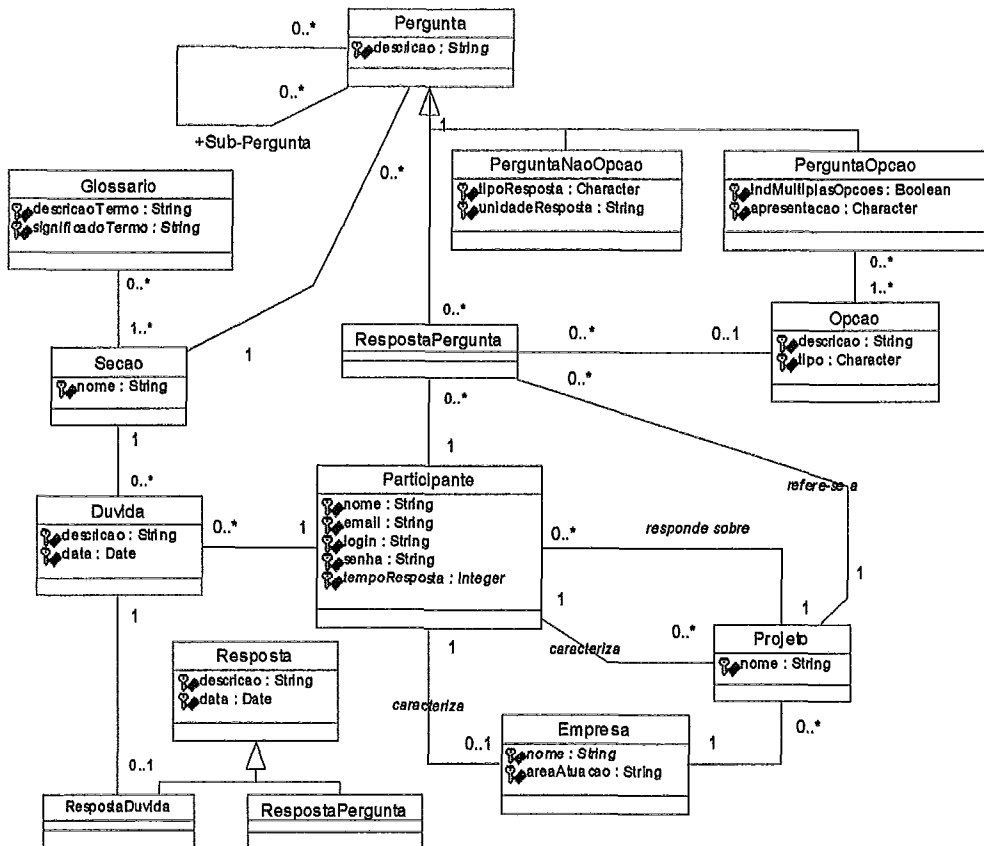


Figura II.5 – Modelo de Classes do Questionário

ANEXO III

**Valores do Teste de Hipóteses e
Análise Descritiva dos Dados**

As tabelas III.1 a III.4 apresentam, para cada grupo de participantes da pesquisa em separado, a proporção de respostas obtidas para cada tipo de conhecimento e os correspondentes valores achados no teste das hipóteses.

A proporção é o número de respostas obtidas para a questão, desconsiderando-se as ocorrências da categoria *Não Avaliado* (ou seja, os casos em que a opção *Não Avaliado* foi assinalada), dividido pelo número total de questionários respondidos pelo grupo. Sobre a proporção, é importante destacar que a mesma diminui para os tipos de conhecimento relacionados a fornecedores, parceiros técnicos e clientes, porque os participantes foram instruídos a não avaliarem estes tipos de conhecimento quando não houvesse fornecedores, parceiros técnicos ou clientes envolvidos nos projetos.

Com relação ao teste das hipóteses, são fornecidos os valores achados do Qui-quadrado (χ^2) e o grau de liberdade (*gl*) da correspondente distribuição de frequência, conforme calculados pelo *software* SPSS v 10.0, além do nível de significância para as hipóteses confirmadas (α). O nível de significância é obtido a partir de uma tabela de valores críticos de Qui-quadrado que encontra-se resumida na tabela III.5 e pode ser encontrada na íntegra em [211]. O valor achado do Qui-quadrado é comparado com os valores esperados da tabela para o mesmo grau de liberdade. A célula em que o valor achado for maior ou igual ao valor fornecido na tabela determina o nível de significância. Hipóteses foram consideradas verdadeiras a partir do nível de significância 0,05.

Tabela III.1 – Valores do Teste de Hipóteses para o Grupo de Programadores

Código	Grupo	Tipo de Conhecimento	Prop. (%)	H1			H2			H3		
				χ^2	<i>gl</i>	α	χ^2	<i>gl</i>	α	χ^2	<i>gl</i>	α
4.01	DM	Domínio da Aplicação	97,6	2,995	4		6,491	6		5,841	6	
5.01.01	ORG	Missão da Organização	100,0	3,917	4		7,842	6		2,256	6	
5.01.02	ORG	Deficiências e Potenciais	100,0	3,928	4		1,970	6		1,391	6	
5.01.03	ORG	Diretrizes e Normas	95,1	7,014	4		5,839	6		2,767	6	
5.01.04	ORG	Objetivos e Metas	95,1	9,213	4		6,890	6		2,079	6	
5.01.05	ORG	Áreas Críticas p/ Objetivos	92,7	0,712	4		5,813	6		1,922	6	
5.01.06	ORG	Produtos e/ou Serviços	90,2	7,047	4		4,188	6		2,041	6	
5.01.07	ORG	Estrutura Organizacional	92,7	1,602	4		8,754	6		16,217	6	0,05
5.01.08	ORG	Alocação dos Profissionais	90,2	1,113	4		8,683	6		8,416	6	
5.01.09	ORG	Distribuição de Competências	92,7	10,500	4	0,05	6,051	6		4,990	6	
5.01.10	ORG	Relações Informais	95,1	1,890	4		3,196	6		10,779	6	
5.01.11	ORG	Processos Organizacionais	92,7	1,869	4		6,619	6		10,331	6	
5.01.12	ORG	Contratos/Acordos	85,4	2,096	4		2,280	4		4,172	6	
6.01.01	OC	Tipos de Software	100,0	7,184	4		13,714	6	0,05	8,914	6	
6.01.02	OC	Processo de Software	95,1	0,117	2		4,116	3		8,093	3	
6.01.03	OC	Avaliação do Processo	95,1	2,424	4		6,341	6		6,716	6	
6.02	OC	Roteiros de Documentos	95,1	5,845	4		12,870	6	0,05	5,021	6	
6.03	OC	Itens de Software	100,0	0,430	2		2,206	3		0,981	3	
6.04.01	MP	Modelagem de Processo	92,7	3,691	4		8,355	6		3,360	6	
6.04.02	MP	Gerência de Projeto	87,8	4,013	4		7,320	6		10,187	6	
6.04.03	MP	Gerência de Riscos	80,5	3,516	4		5,549	6		8,621	6	
6.04.04	MP	Gerência de Qualidade	85,4	4,060	4		7,663	6		2,604	6	
6.04.05	MP	Gerência de Configuração	85,4	2,815	4		4,662	6		10,184	6	
6.04.06	MP	Documentação	97,6	1,327	4		21,542	6	0,01	5,636	6	
6.04.07	MP	Engenharia de Sistemas	82,9	1,306	4		4,812	6		10,871	6	
6.04.08	MP	Análise de Requisitos	92,7	8,701	4		6,059	6		5,097	6	
6.04.09	MP	Projeto (design)	97,6	2,883	4		9,022	6		7,997	6	
6.04.10	MP	Codificação	100,0	3,826	4		3,425	6		8,110	6	
6.04.11	MP	Testes	100,0	3,304	4		3,925	6		6,762	6	

Tabela III.1 – Valores do Teste de Hipóteses para o Grupo de Programadores (cont.)

Código	Grupo	Tipo de Conhecimento	Prop. (%)	H1			H2			H3		
				χ^2	gl	α	χ^2	gl	α	χ^2	gl	α
6.04.12	MP	Operação/Suporte a Usuários	97,6	2,582	4		8,351	6		8,864	6	
6.04.13	MP	Manutenção	97,6	6,073	4		7,965	6		5,959	6	
6.04.14	MP	Engenharia Reversa e Reeng	90,2	6,339	4		7,614	6		1,679	6	
6.04.15	MP	Reuso de Itens de Software	100,0	4,111	4		6,650	6		3,650	6	
6.04.16	MP	Melhoria de Processo	92,7	1,708	2		3,711	3		1,213	3	
6.04.17	MP	Treinamento	95,1	7,677	4		7,875	6		4,581	6	
6.04.18	MP	Processo de Aquisição	85,4	2,002	4		6,149	6		2,386	6	
6.04.19	MP	Processo de Fornecimento	80,5	1,071	4		4,307	6		1,957	6	
6.04.20	MP	Solução de Problemas	92,7	3,278	4		4,774	6		6,413	6	
6.05.01	LA	Modelagem de Processo	80,5	3,986	4		4,641	6		5,172	6	
6.05.02	LA	Gerência de Projeto	82,9	5,050	4		5,211	6		6,409	6	
6.05.03	LA	Gerência de Riscos	78,0	8,820	4		5,025	6		6,438	6	
6.05.04	LA	Gerência de Qualidade	82,9	3,011	4		8,653	6		5,910	6	
6.05.05	LA	Gerência de Configuração	80,5	4,053	4		10,903	6		4,726	6	
6.05.06	LA	Documentação	92,7	4,365	4		1,988	6		3,898	6	
6.05.07	LA	Engenharia de Sistemas	82,9	5,698	4		5,257	6		2,153	6	
6.05.08	LA	Análise de Requisitos	97,6	7,693	4		4,681	6		3,113	6	
6.05.09	LA	Projeto (design)	97,6	2,737	4		6,003	6		10,415	6	
6.05.10	LA	Codificação	100,0	1,440	2		4,703	3		1,119	3	
6.05.11	LA	Testes	100,0	7,410	4		2,699	6		4,724	6	
6.05.12	LA	Operação/Suporte a Usuários	95,1	2,443	4		12,584	6		8,504	6	
6.05.13	LA	Manutenção	95,1	7,020	4		22,469	6	0,01	4,673	6	
6.05.14	LA	Engenharia Reversa e Reeng	85,4	1,330	4		5,610	6		4,820	6	
6.05.15	LA	Reuso de Itens de Software	100,0	2,917	4		5,250	6		2,298	6	
6.05.16	LA	Melhoria de Processo	92,7	5,312	4		5,073	6		6,084	6	
6.05.17	LA	Treinamento	90,2	4,228	4		11,929	6		6,678	6	
6.05.18	LA	Processo de Aquisição	85,4	6,553	4		5,202	6		3,963	6	
6.05.19	LA	Processo de Fornecimento	73,2	5,167	4		9,353	6		2,510	6	
6.05.20	LA	Solução de Problemas	92,7	0,974	4		4,484	6		4,168	6	
6.06.01	DH	Gerência de Projeto	82,9	3,546	4		5,910	4		14,107	6	0,05
6.06.02	DH	Gerência de Riscos	85,4	15,058	4	0,01	6,421	4		7,732	6	
6.06.03	DH	Gerência de Qualidade	85,4	5,608	4		7,455	4		8,628	6	
6.06.04	DH	Treinamento	85,4	3,487	4		5,667	6		10,313	6	
6.06.05	DH	Métricas de Produto	87,8	6,202	4		5,065	6		9,121	6	
6.06.06	DH	Métricas de Processo	87,8	6,090	4		9,191	6		5,164	6	
6.07	OC	Respostas para Perguntas	95,1	3,330	4		5,565	6		7,489	6	
7.01.01	FN	Missão da Organização	82,9	4,128	4		4,000	6		9,294	6	
7.01.02	FN	Deficiências e Potenciais	82,9	2,520	4		7,293	6		3,314	6	
7.01.03	FN	Diretrizes e Normas	82,9	5,126	4		5,323	6		3,872	6	
7.01.04	FN	Produtos e/ou Serviços	85,4	3,778	4		3,196	6		10,365	6	
7.01.05	FN	Estrutura Organizacional	82,9	0,284	4		4,506	6		4,484	6	
7.01.06	FN	Alocação dos Profissionais	80,5	2,969	4		6,312	6		9,667	6	
7.01.07	FN	Distribuição de Competências	82,9	0,984	4		2,937	6		6,415	6	
7.01.08	FN	Relações Informais	80,5	0,934	4		5,323	6		3,813	6	
7.01.09	FN	Processos Organizacionais	80,5	3,044	4		2,259	6		3,152	6	
8.01.01	PT	Missão da Organização	73,2	1,897	4		6,530	6		10,961	6	
8.01.02	PT	Deficiências e Potenciais	73,2	2,886	4		1,512	6		12,651	6	0,05
8.01.03	PT	Diretrizes e Normas	70,7	0,706	4		4,461	6		10,407	6	
8.01.04	PT	Produtos e/ou Serviços	70,7	3,019	4		6,443	6		13,401	6	0,05
8.01.05	PT	Estrutura Organizacional	73,2	1,401	4		3,240	6		8,950	6	
8.01.06	PT	Alocação dos Profissionais	73,2	2,322	4		2,962	6		6,520	6	
8.01.07	PT	Distribuição de Competências	73,2	1,295	4		4,501	6		5,515	6	
8.01.08	PT	Relações Informais	73,2	3,835	4		9,411	6		6,330	6	
8.01.09	PT	Processos Organizacionais	73,2	0,564	4		4,714	6		5,187	6	
9.01.01	CL	Missão da Organização	73,2	3,954	4		6,374	4		6,813	6	
9.01.02	CL	Deficiências e Potenciais	75,6	4,906	4		12,639	6	0,05	1,885	6	
9.01.03	CL	Diretrizes e Normas	75,6	9,343	4		15,488	6	0,05	10,923	6	
9.01.04	CL	Produtos e/ou Serviços	75,6	5,501	4		14,513	6	0,05	3,160	6	
9.01.05	CL	Estrutura Organizacional	75,6	2,669	4		11,654	6		9,230	6	
9.01.06	CL	Alocação dos Profissionais	75,6	3,040	4		7,655	6		5,494	6	
9.01.07	CL	Distribuição de Competências	73,2	4,166	4		6,873	6		8,494	6	
9.01.08	CL	Relações Informais	70,7	0,660	4		8,567	4		8,228	6	
9.01.09	CL	Processos Organizacionais	73,2	3,377	4		5,207	6		5,086	6	
10.01.01	LT	Conhecimento Teórico em ES	87,8	2,888	4		4,819	6		12,537	6	
10.01.02	LT	Inovações Tecnológicas	87,8	4,662	4		3,341	6		7,162	6	
10.01.03	LT	LA da Indústria de Software	87,8	3,047	4		3,192	6		3,622	6	
10.01.04	LT	MP da Indústria de Software	87,8	6,186	4		4,326	6		3,140	6	

Tabela III.2 – Valores do Teste de Hipóteses para o Grupo de Analistas

Código	Grupo	Tipo de Conhecimento	Prop. (%)	H1			H2			H3		
				χ^2	gl	α	χ^2	gl	α	χ^2	gl	α
4.01	DM	Domínio da Aplicação	98,6	3,461	4		9,494	6		4,254	6	
5.01.01	ORG	Missão da Organização	100,0	2,508	4		8,167	6		8,574	6	
5.01.02	ORG	Deficiências e Potenciais	95,7	1,610	4		17,109	6	0,01	8,689	6	
5.01.03	ORG	Diretrizes e Normas	100,0	1,528	4		6,975	6		9,361	6	
5.01.04	ORG	Objetivos e Metas	100,0	3,012	4		5,241	6		11,024	6	
5.01.05	ORG	Áreas Críticas p/ Objetivos	100,0	1,672	4		11,097	6		11,014	6	
5.01.06	ORG	Produtos e/ou Serviços	98,6	1,005	4		19,219	6	0,01	6,064	6	
5.01.07	ORG	Estrutura Organizacional	98,6	0,365	4		10,102	6		7,315	6	
5.01.08	ORG	Alocação dos Profissionais	95,7	4,035	4		9,516	6		4,263	6	
5.01.09	ORG	Distribuição de Competências	97,1	0,508	4		12,546	6		2,607	6	
5.01.10	ORG	Relações Informais	94,3	0,704	4		7,132	6		6,858	6	
5.01.11	ORG	Processos Organizacionais	100,0	1,010	4		21,818	6	0,01	6,423	6	
5.01.12	ORG	Contratos/Acordos	94,3	1,260	4		4,101	6		9,468	6	
6.01.01	OC	Tipos de Software	100,0	2,681	4		6,494	6		11,682	6	
6.01.02	OC	Processo de Software	98,6	6,275	4		6,887	6		4,092	6	
6.01.03	OC	Avaliação do Processo	97,1	0,511	2		3,690	3		3,651	3	
6.02	OC	Roteiros de Documentos	98,6	3,080	4		5,296	6		4,814	6	
6.03	OC	Ítems de Software	100,0	7,768	4		3,952	6		8,836	6	
6.04.01	MP	Modelagem de Processo	97,1	4,257	4		21,600	6	0,01	3,307	6	
6.04.02	MP	Gerência de Projeto	98,6	3,904	4		7,391	6		3,638	6	
6.04.03	MP	Gerência de Riscos	94,3	8,875	4		11,259	6		1,777	6	
6.04.04	MP	Gerência de Qualidade	95,7	5,169	4		8,979	6		4,620	6	
6.04.05	MP	Gerência de Configuração	91,4	8,646	4		4,515	6		8,676	6	
6.04.06	MP	Documentação	98,6	3,065	4		5,492	6		5,262	6	
6.04.07	MP	Engenharia de Sistemas	95,7	1,458	4		4,900	6		3,425	6	
6.04.08	MP	Análise de Requisitos	97,1	0,543	2		5,321	3		1,015	3	
6.04.09	MP	Projeto (design)	97,1	1,812	4		7,684	6		5,958	6	
6.04.10	MP	Codificação	100,0	4,357	4		3,989	6		3,756	6	
6.04.11	MP	Testes	100,0	3,434	2		2,633	3		1,756	3	
6.04.12	MP	Operação/Suporte a Usuários	98,6	3,475	4		5,129	6		11,274	6	
6.04.13	MP	Manutenção	100,0	3,231	4		6,928	6		6,128	6	
6.04.14	MP	Engenharia Reversa e Reeng	90,0	7,123	4		15,025	6		14,212	6	0,05
6.04.15	MP	Reuso de Ítems de Software	95,7	6,429	4		4,687	6		12,387	6	
6.04.16	MP	Melhoria de Processo	97,1	2,099	4		2,950	6		18,388	6	0,01
6.04.17	MP	Treinamento	97,1	2,281	4		9,863	6		3,051	6	
6.04.18	MP	Processo de Aquisição	85,7	2,321	4		4,639	6		6,703	6	
6.04.19	MP	Processo de Fornecimento	78,6	0,759	4		8,565	6		10,580	6	
6.04.20	MP	Solução de Problemas	97,1	3,479	2		1,408	3		13,938	6	0,05
6.05.01	LA	Modelagem de Processo	95,7	2,555	4		10,351	6		8,307	6	
6.05.02	LA	Gerência de Projeto	94,3	1,955	4		37,455	6	0,01	5,211	6	
6.05.03	LA	Gerência de Riscos	92,9	6,556	4		5,863	6		16,015	6	0,05
6.05.04	LA	Gerência de Qualidade	92,9	1,633	4		3,014	6		13,612	6	0,05
6.05.05	LA	Gerência de Configuração	90	11,054	4	0,05	5,882	6		8,631	6	
6.05.06	LA	Documentação	97,1	4,731	4		9,577	6		8,837	6	
6.05.07	LA	Engenharia de Sistemas	92,9	4,034	4		5,262	6		7,933	6	
6.05.08	LA	Análise de Requisitos	97,1	3,266	4		3,498	6		10,721	6	
6.05.09	LA	Projeto (design)	95,7	0,724	4		7,535	6		5,631	6	
6.05.10	LA	Codificação	98,6	5,610	4		8,483	6		6,351	6	
6.05.11	LA	Testes	98,6	1,519	2		7,378	3		1,884	3	
6.05.12	LA	Operação/Suporte a Usuários	97,1	5,391	4		5,754	6		9,829	6	
6.05.13	LA	Manutenção	95,7	5,396	4		3,269	6		4,055	6	
6.05.14	LA	Engenharia Reversa e Reeng	87,1	4,491	4		10,998	6		4,951	6	
6.05.15	LA	Reuso de Ítems de Software	97,1	1,751	4		4,835	6		7,156	6	
6.05.16	LA	Melhoria de Processo	95,7	5,201	4		15,389	6	0,05	6,400	6	
6.05.17	LA	Treinamento	97,1	3,171	4		8,499	6		2,202	6	
6.05.18	LA	Processo de Aquisição	81,4	0,578	4		2,829	6		8,980	6	
6.05.19	LA	Processo de Fornecimento	81,4	3,270	4		9,154	6		5,828	6	
6.05.20	LA	Solução de Problemas	95,7	5,069	4		8,003	6		2,139	3	
6.06.01	DH	Gerência de Projeto	97,1	6,480	4		14,651	6	0,05	7,648	6	
6.06.02	DH	Gerência de Riscos	92,9	1,596	4		6,691	6		6,753	6	
6.06.03	DH	Gerência de Qualidade	94,3	2,386	4		15,057	6	0,05	5,828	6	
6.06.04	DH	Treinamento	97,1	4,557	4		9,467	6		8,734	6	
6.06.05	DH	Métricas de Produto	97,1	6,558	4		10,132	6		7,831	6	
6.06.06	DH	Métricas de Processo	94,3	6,698	4		7,937	6		5,940	6	
6.07	OC	Respostas para Perguntas	94,3	3,565	4		9,872	6		7,674	4	
7.01.01	FN	Missão da Organização	78,6	1,669	4		9,040	6		7,391	6	
7.01.02	FN	Deficiências e Potenciais	80,0	2,008	4		3,800	6		13,224	6	0,05

Tabela III.2 – Valores do Teste de Hipóteses para o Grupo de Analistas (cont.)

Código	Grupo	Tipo de Conhecimento	Prop. (%)	H1			H2			H3		
				χ^2	gl	α	χ^2	gl	α	χ^2	gl	α
7.01.03	FN	Diretrizes e Normas	78,6	0,827	4		4,981	6		6,928	6	
7.01.04	FN	Produtos e/ou Serviços	78,6	1,183	4		3,189	6		11,082	6	
7.01.05	FN	Estrutura Organizacional	81,4	1,697	4		5,539	6		3,900	6	
7.01.06	FN	Alocação dos Profissionais	80,0	6,012	4		6,758	6		1,790	6	
7.01.07	FN	Distribuição de Competências	80,0	4,891	4		11,661	6		8,822	6	
7.01.08	FN	Relações Informais	80,0	4,422	4		2,812	6		5,858	6	
7.01.09	FN	Processos Organizacionais	80,0	1,768	4		4,706	6		9,701	6	
8.01.01	PT	Missão da Organização	77,1	2,475	4		9,348	6		5,357	6	
8.01.02	PT	Deficiências e Potenciais	77,1	3,778	4		2,707	6		9,324	6	
8.01.03	PT	Diretrizes e Normas	77,1	0,804	4		4,496	6		1,633	6	
8.01.04	PT	Produtos e/ou Serviços	77,1	12,419	4	0,05	2,738	6		5,906	6	
8.01.05	PT	Estrutura Organizacional	80,0	2,423	4		4,983	6		8,539	6	
8.01.06	PT	Alocação dos Profissionais	80,0	11,707	4	0,05	9,563	6		8,231	6	
8.01.07	PT	Distribuição de Competências	78,6	1,061	4		7,659	6		11,712	6	
8.01.08	PT	Relações Informais	78,6	1,412	4		6,388	6		4,449	6	
8.01.09	PT	Processos Organizacionais	78,6	8,190	4		5,137	6		5,619	6	
9.01.01	CL	Missão da Organização	85,7	0,887	4		9,095	6		8,035	6	
9.01.02	CL	Deficiências e Potenciais	87,1	1,457	4		5,368	6		3,388	6	
9.01.03	CL	Diretrizes e Normas	85,7	2,882	4		11,458	6		18,765	6	0,01
9.01.04	CL	Produtos e/ou Serviços	87,1	2,348	4		8,738	6		7,327	6	
9.01.05	CL	Estrutura Organizacional	88,6	5,281	4		2,755	6		10,979	6	
9.01.06	CL	Alocação dos Profissionais	88,6	1,366	4		2,878	6		15,791	6	0,05
9.01.07	CL	Distribuição de Competências	88,6	6,937	4		7,698	6		11,304	6	
9.01.08	CL	Relações Informais	84,3	1,727	4		8,852	6		9,910	6	
9.01.09	CL	Processos Organizacionais	88,6	0,858	4		15,851	6	0,05	17,217	6	0,01
10.01.01	LT	Conhecimento Teórico em ES	98,6	5,118	4		18,222	6	0,01	8,598	6	
10.01.02	LT	Inovações Tecnológicas	98,6	5,419	4		14,086	6	0,05	8,685	6	
10.01.03	LT	LA da Indústria de Software	98,6	6,763	4		9,668	6		6,491	6	
10.01.04	LT	MP da Indústria de Software	98,6	4,856	4		35,073	6	0,01	6,578	6	

Tabela III.3 – Valores do Teste de Hipóteses para o Grupo de Gerentes

Código	Grupo	Tipo de Conhecimento	Prop. (%)	H1			H2			H3		
				χ^2	gl	α	χ^2	gl	α	χ^2	gl	α
4.01	DM	Domínio da Aplicação	100,0	3,249	4		4,457	6		2,453	6	
5.01.01	ORG	Missão da Organização	96,9	3,638	4		9,192	6		2,891	6	
5.01.02	ORG	Deficiências e Potenciais	96,9	1,536	4		7,107	6		14,919	6	0,05
5.01.03	ORG	Diretrizes e Normas	100,0	1,372	4		5,726	6		11,596	6	
5.01.04	ORG	Objetivos e Metas	100,0	0,890	4		6,014	6		20,927	6	0,01
5.01.05	ORG	Áreas Críticas p/ Objetivos	96,9	2,443	4		3,546	6		12,416	6	
5.01.06	ORG	Produtos e/ou Serviços	96,9	7,176	4		2,343	6		7,127	6	
5.01.07	ORG	Estrutura Organizacional	100,0	11,218	4	0,05	4,362	6		5,659	6	
5.01.08	ORG	Alocação dos Profissionais	100,0	8,885	4		5,936	6		17,012	6	0,01
5.01.09	ORG	Distribuição de Competências	100,0	6,230	4		4,646	6		18,563	6	0,01
5.01.10	ORG	Relações Informais	100,0	1,118	4		4,735	6		7,152	6	
5.01.11	ORG	Processos Organizacionais	100,0	3,452	2		3,308	3		2,763	3	
5.01.12	ORG	Contratos/Acordos	96,9	3,656	4		7,417	6		13,060	6	0,05
6.01.01	OC	Tipos de Software	100,0	5,607	4		3,600	6		6,478	6	
6.01.02	OC	Processo de Software	100,0	3,746	4		3,615	6		4,236	6	
6.01.03	OC	Avaliação do Processo	100,0	3,685	4		7,552	6		8,396	6	
6.02	OC	Roteiros de Documentos	100,0	2,795	4		4,985	6		6,826	6	
6.03	OC	Ítems de Software	100,0	0,371	2		5,156	3		0,489	3	
6.04.01	MP	Modelagem de Processo	100,0	2,169	2		8,000	3		2,109	3	
6.04.02	MP	Gerência de Projeto	100,0	0,980	2		1,195	3		4,775	3	
6.04.03	MP	Gerência de Riscos	100,0	3,047	2		4,666	6		11,797	6	
6.04.04	MP	Gerência de Qualidade	100,0	0,564	2		1,827	3		13,224	3	0,01
6.04.05	MP	Gerência de Configuração	96,9	1,841	4		7,900	6		8,507	6	
6.04.06	MP	Documentação	100,0	1,786	2		3,174	3		18,061	3	0,01
6.04.07	MP	Engenharia de Sistemas	100,0	2,639	2		3,174	3		9,370	3	0,05
6.04.08	MP	Análise de Requisitos	100,0	2,049	2		2,486	3		2,109	3	
6.04.09	MP	Projeto (design)	100,0	3,464	2		3,174	3		2,668	3	
6.04.10	MP	Codificação	100,0	4,746	4		1,612	6		12,447	6	
6.04.11	MP	Testes	100,0	5,272	2		1,827	3		13,453	3	0,01
6.04.12	MP	Operação/Suporte a Usuários	100,0	2,203	4		3,744	6		11,560	6	
6.04.13	MP	Manutenção	96,9	5,584	4		4,060	6		6,886	6	
6.04.14	MP	Engenharia Reversa e Reeng	96,9	1,278	4		2,942	6		12,549	6	

Tabela III.3 – Valores do Teste de Hipóteses para o Grupo de Gerentes (cont.)

Código	Grupo	Tipo de Conhecimento	Prop. (%)	H1			H2			H3		
				χ^2	gl	α	χ^2	gl	α	χ^2	gl	α
6.04.15	MP	Reuso de Itens de Software	100,0	2,951	4		16,815	6	0,01	9,580	6	
6.04.16	MP	Melhoria de Processo	100,0	4,813	4		1,720	6		3,674	6	
6.04.17	MP	Treinamento	100,0	9,029	4		3,743	6		4,308	6	
6.04.18	MP	Processo de Aquisição	96,9	3,126	4		7,902	6		5,542	6	
6.04.19	MP	Processo de Fornecimento	100,0	7,981	4		3,712	6		4,372	6	
6.04.20	MP	Solução de Problemas	100,0	5,111	4		4,646	6		6,800	6	
6.05.01	LA	Modelagem de Processo	100,0	4,426	4		2,432	6		6,826	6	
6.05.02	LA	Gerência de Projeto	100,0	1,258	2		3,549	3		3,612	3	
6.05.03	LA	Gerência de Riscos	96,9	9,248	4		3,279	6		14,174	6	0,05
6.05.04	LA	Gerência de Qualidade	100,0	5,473	4		5,066	6		9,226	6	
6.05.05	LA	Gerência de Configuração	93,8	2,157	4		5,394	6		13,348	6	0,05
6.05.06	LA	Documentação	100,0	0,693	2		1,962	3		10,430	3	0,05
6.05.07	LA	Engenharia de Sistemas	100,0	3,386	4		6,010	6		8,235	6	
6.05.08	LA	Análise de Requisitos	100,0	0,175	2		4,699	3		4,563	3	
6.05.09	LA	Projeto (design)	100,0	0,659	2		5,222	3		4,790	3	
6.05.10	LA	Codificação	100,0	5,702	4		2,948	6		4,737	6	
6.05.11	LA	Testes	100,0	5,246	4		3,615	6		5,008	6	
6.05.12	LA	Operação/Suporte a Usuários	100,0	2,062	4		4,306	6		6,371	6	
6.05.13	LA	Manutenção	96,9	3,054	4		8,283	6		7,388	6	
6.05.14	LA	Engenharia Reversa e Reeng	93,8	2,566	4		4,078	6		11,329	6	
6.05.15	LA	Reuso de Itens de Software	96,9	2,354	4		15,821	6	0,05	10,046	6	
6.05.16	LA	Melhoria de Processo	96,9	2,334	4		7,186	6		6,965	6	
6.05.17	LA	Treinamento	100,0	4,594	4		4,705	6		9,167	6	
6.05.18	LA	Processo de Aquisição	96,9	4,802	4		4,981	6		5,865	6	
6.05.19	LA	Processo de Fornecimento	96,9	2,531	4		4,576	6		3,082	6	
6.05.20	LA	Solução de Problemas	100,0	0,212	2		5,810	3		4,695	3	
6.06.01	DH	Gerência de Projeto	100,0	5,695	4		3,843	6		3,174	6	
6.06.02	DH	Gerência de Riscos	100,0	7,973	4		4,950	6		3,258	6	
6.06.03	DH	Gerência de Qualidade	100,0	0,028	2		3,549	3		0,707	3	
6.06.04	DH	Treinamento	100,0	8,603	4		3,086	6		1,858	6	
6.06.05	DH	Métricas de Produto	100,0	4,560	4		2,580	6		4,924	6	
6.06.06	DH	Métricas de Processo	100,0	2,493	4		2,926	6		10,335	6	
6.07	OC	Respostas para Perguntas	100,0	0,659	2		1,420	3		1,705	3	
7.01.01	FN	Missão da Organização	81,3	1,830	4		3,783	6		8,168	6	
7.01.02	FN	Deficiências e Potenciais	81,3	1,475	4		4,940	6		8,673	6	
7.01.03	FN	Diretrizes e Normas	81,3	1,354	4		9,232	6		7,783	6	
7.01.04	FN	Produtos e/ou Serviços	84,4	4,779	4		3,664	6		8,861	6	
7.01.05	FN	Estrutura Organizacional	81,3	6,007	4		11,552	6		8,811	6	
7.01.06	FN	Alocação dos Profissionais	81,3	3,857	4		8,346	6		7,654	6	
7.01.07	FN	Distribuição de Competências	81,3	1,416	4		7,274	6		8,352	6	
7.01.08	FN	Relações Informais	81,3	1,714	4		12,675	6	0,05	7,120	6	
7.01.09	FN	Processos Organizacionais	81,3	9,429	4		8,901	6		9,890	6	
8.01.01	PT	Missão da Organização	75,0	3,022	4		4,456	6		6,050	6	
8.01.02	PT	Deficiências e Potenciais	75,0	3,548	4		4,866	6		6,263	6	
8.01.03	PT	Diretrizes e Normas	75,0	3,915	4		7,947	6		4,170	6	
8.01.04	PT	Produtos e/ou Serviços	75,0	9,921	4	0,05	9,070	6		5,136	6	
8.01.05	PT	Estrutura Organizacional	75,0	4,997	4		7,532	6		8,897	6	
8.01.06	PT	Alocação dos Profissionais	75,0	4,749	4		6,236	6		10,374	6	
8.01.07	PT	Distribuição de Competências	75,0	9,528	4	0,05	7,165	6		9,608	6	
8.01.08	PT	Relações Informais	75,0	4,260	4		6,514	6		9,103	6	
8.01.09	PT	Processos Organizacionais	75,0	5,503	4		7,947	6		6,804	6	
9.01.01	CL	Missão da Organização	90,6	4,389	4		8,642	6		3,493	6	
9.01.02	CL	Deficiências e Potenciais	90,6	5,780	4		11,969	6		6,065	6	
9.01.03	CL	Diretrizes e Normas	90,6	2,699	4		7,398	6		11,981	6	
9.01.04	CL	Produtos e/ou Serviços	90,6	8,468	4		8,815	6		10,287	6	
9.01.05	CL	Estrutura Organizacional	90,6	15,377	4	0,01	10,018	6		3,113	6	
9.01.06	CL	Alocação dos Profissionais	90,6	5,494	4		5,558	6		3,722	6	
9.01.07	CL	Distribuição de Competências	90,6	3,418	4		6,821	6		13,747	6	0,05
9.01.08	CL	Relações Informais	90,6	2,576	4		5,207	6		2,769	6	
9.01.09	CL	Processos Organizacionais	90,6	3,481	4		6,620	6		7,965	6	
10.01.01	LT	Conhecimento Teórico em ES	100,0	2,498	4		1,612	6		3,853	6	
10.01.02	LT	Inovações Tecnológicas	100,0	0,269	2		3,174	3		1,356	3	
10.01.03	LT	LA da Indústria de Software	100,0	1,217	2		2,486	3		5,746	3	
10.01.04	LT	MP da Indústria de Software	100,0	1,217	2		2,486	3		5,746	3	

Tabela III.4 – Valores do Teste de Hipóteses para o Grupo de Diretores

Código	Grupo	Tipo de Conhecimento	Prop. (%)	H1			H2			H3		
				χ^2	gl	α	χ^2	gl	α	χ^2	gl	α
4.01	DM	Domínio da Aplicação	90,0	*	*	*	*	*	*	*	*	*
5.01.01	ORG	Missão da Organização	90,0	4,250	4		3,139	4		1,046	2	
5.01.02	ORG	Deficiências e Potenciais	90,0	4,186	4		5,912	4		3,819	2	
5.01.03	ORG	Diretrizes e Normas	90,0	6,820	4		3,001	4		1,995	2	
5.01.04	ORG	Objetivos e Metas	90,0	6,959	4		4,186	4		3,819	2	
5.01.05	ORG	Áreas Críticas p/ Objetivos	90,0	4,186	4		4,186	4		3,819	2	
5.01.06	ORG	Produtos e/ou Serviços	90,0	2,863	2		0,228	2		0,228	1	
5.01.07	ORG	Estrutura Organizacional	80,0	8,997	4		4,132	4		6,766	2	0,05
5.01.08	ORG	Alocação dos Profissionais	80,0	3,452	4		4,132	4		0,178	2	
5.01.09	ORG	Distribuição de Competências	100,0	1,498	2		2,683	2		2,683	1	
5.01.10	ORG	Relações Informais	90,0	6,820	4		4,589	4		0,908	2	
5.01.11	ORG	Processos Organizacionais	90,0	9,593	4	0,05	5,774	4		1,995	2	
5.01.12	ORG	Contratos/Acordos	90,0	3,001	4		4,589	4		0,908	2	
6.01.01	OC	Tipos de Software	100,0	6,189	4		3,958	4		1,632	2	
6.01.02	OC	Processo de Software	100,0	1,185	2		1,185	2		1,632	1	
6.01.03	OC	Avaliação do Processo	100,0	0,635	2		1,668	2		2,657	1	
6.02	OC	Roteiros de Documentos	100,0	4,441	4		5,487	4		2,657	2	
6.03	OC	Itens de Software	100,0	9,445	4		5,487	4		2,657	2	
6.04.01	MP	Modelagem de Processo	100,0	2,231	2		3,278	2		1,632	1	
6.04.02	MP	Gerência de Projeto	100,0	4,444	2		1,498	2		0,760	1	
6.04.03	MP	Gerência de Riscos	90,0	3,938	2		1,780	2		0,872	1	
6.04.04	MP	Gerência de Qualidade	100,0	4,444	2		1,498	2		0,760	1	
6.04.05	MP	Gerência de Configuração	90,0	9,535	2	0,01	3,990	2		1,897	1	
6.04.06	MP	Documentação	100,0	2,231	2		3,278	2		1,632	1	
6.04.07	MP	Engenharia de Sistemas	100,0	4,444	2		1,498	2		0,760	1	
6.04.08	MP	Análise de Requisitos	100,0	9,445	4		2,715	4		2,657	2	
6.04.09	MP	Projeto (design)	100,0	2,231	2		3,278	2		1,632	1	
6.04.10	MP	Codificação	100,0	2,683	2		1,498	2		0,760	1	
6.04.11	MP	Testes	100,0	1,498	2		1,498	2		0,760	1	
6.04.12	MP	Operação/Suporte a Usuários	100,0	1,498	2		1,498	2		0,760	1	
6.04.13	MP	Manutenção	100,0	8,260	4		5,487	4		2,657	2	
6.04.14	MP	Engenharia Reversa e Reeng	90,0	5,774	4		6,820	4		0,375	2	
6.04.15	MP	Reuso de Itens de Software	100,0	9,445	4		5,487	4		2,657	2	
6.04.16	MP	Melhoria de Processo	100,0	1,185	2		3,278	2		1,632	1	
6.04.17	MP	Treinamento	100,0	3,958	4		3,278	4		1,632	2	
6.04.18	MP	Processo de Aquisição	90,0	4,186	4		3,139	4		1,046	2	
6.04.19	MP	Processo de Fornecimento	100,0	3,278	2		3,416	2		6,189	1	0,05
6.04.20	MP	Solução de Problemas	100,0	3,416	2		3,278	2		1,632	1	
6.05.01	LA	Modelagem de Processo	90,0	8,685	4		6,959	4		3,139	2	
6.05.02	LA	Gerência de Projeto	100,0	3,958	4		3,278	4		1,632	2	
6.05.03	LA	Gerência de Riscos	90,0	13,550	4	0,01	9,052	4		6,959	2	0,05
6.05.04	LA	Gerência de Qualidade	100,0	4,637	4		3,591	4		0,760	2	
6.05.05	LA	Gerência de Configuração	90,0	6,000	4		3,990	4		1,897	2	
6.05.06	LA	Documentação	100,0	4,441	4		5,487	4		2,657	2	
6.05.07	LA	Engenharia de Sistemas	100,0	9,445	4		5,487	4		2,657	2	
6.05.08	LA	Análise de Requisitos	100,0	10,008	4	0,05	3,278	4		1,632	2	
6.05.09	LA	Projeto (design)	100,0	5,684	4		4,637	4		1,807	2	
6.05.10	LA	Codificação	100,0	5,626	2		5,487	2		2,657	1	
6.05.11	LA	Testes	100,0	1,498	2		1,498	2		0,760	1	
6.05.12	LA	Operação/Suporte a Usuários	100,0	4,441	4		5,487	4		2,657	2	
6.05.13	LA	Manutenção	100,0	7,410	4		8,456	4		3,900	2	
6.05.14	LA	Engenharia Reversa e Reeng	90,0	11,319	4	0,05	12,365	4	0,05	6,959	2	0,05
6.05.15	LA	Reuso de Itens de Software	100,0	9,641	4	0,05	8,456	4		3,900	2	
6.05.16	LA	Melhoria de Processo	100,0	7,271	4		13,863	4	0,01	5,487	2	
6.05.17	LA	Treinamento	100,0	2,773	4		5,545	4		0,988	2	
6.05.18	LA	Processo de Aquisição	90,0	3,139	2		2,093	2		0,000	1	
6.05.19	LA	Processo de Fornecimento	100,0	6,672	4		5,487	4		2,657	2	
6.05.20	LA	Solução de Problemas	100,0	2,683	2		1,498	2		0,760	1	
6.06.01	DH	Gerência de Projeto	100,0	1,498	2		1,498	2		0,760	1	
6.06.02	DH	Gerência de Riscos	90,0	2,460	2		1,780	2		0,872	1	
6.06.03	DH	Gerência de Qualidade	100,0	2,231	2		3,278	2		1,832	1	
6.06.04	DH	Treinamento	100,0	3,958	4		3,278	4		1,632	2	
6.06.05	DH	Métricas de Produto	100,0	10,008	4	0,05	3,278	4		1,632	2	
6.06.06	DH	Métricas de Processo	100,0	2,683	2		1,498	2		0,760	1	
6.07	OC	Respostas para Perguntas	100,0	2,683	2		1,498	2		0,760	1	
7.01.01	FN	Missão da Organização	80,0	5,178	4		4,499	4		0,680	2	
7.01.02	FN	Deficiências e Potenciais	80,0	2,406	4		4,499	4		0,680	2	

Tabela III.4 – Valores do Teste de Hipóteses para o Grupo de Diretores (cont.)

Código	Grupo	Tipo de Conhecimento	Prop. (%)	H1		H2		H3		
7.01.03	FN	Diretrizes e Normas	80,0	1,359	4	3,452	4	2,406	2	
7.01.04	FN	Produtos e/ou Serviços	80,0	6,766	2	0,05	2,267	2	0,178	1
7.01.05	FN	Estrutura Organizacional	80,0	0,680	2	3,819	2	0,000	1	
7.01.06	FN	Alocação dos Profissionais	80,0	6,904	4	7,271	4	8,997	2	0,05
7.01.07	FN	Distribuição de Competências	80,0	3,452	4	5,545	4	1,726	2	
7.01.08	FN	Relações Informais	80,0	3,993	4	7,133	4	3,993	2	
7.01.09	FN	Processos Organizacionais	80,0	1,359	4	6,225	4	2,406	2	
8.01.01	PT	Missão da Organização	90,0	3,139	2	1,955	2	1,897	1	
8.01.02	PT	Deficiências e Potenciais	90,0	1,780	2	1,275	2	0,537	1	
8.01.03	PT	Diretrizes e Normas	90,0	2,263	2	2,805	2	1,159	1	
8.01.04	PT	Produtos e/ou Serviços	90,0	9,535	2	0,01	2,805	2	1,159	1
8.01.05	PT	Estrutura Organizacional	90,0	4,048	4	7,361	4	2,805	2	
8.01.06	PT	Alocação dos Profissionais	90,0	4,186	4	4,727	4	1,897	2	
8.01.07	PT	Distribuição de Competências	90,0	1,780	2	1,275	2	0,537	1	
8.01.08	PT	Relações Informais	90,0	5,774	4	7,361	4	2,805	2	
8.01.09	PT	Processos Organizacionais	90,0	4,250	4	4,727	4	1,897	2	
9.01.01	CL	Missão da Organização	100,0	5,487	4	4,441	4	6,476	2	0,05
9.01.02	CL	Deficiências e Potenciais	100,0	0,541	2	0,541	2	0,483	1	
9.01.03	CL	Diretrizes e Normas	100,0	1,668	2	2,715	2	0,023	1	
9.01.04	CL	Produtos e/ou Serviços	100,0	4,441	4	3,761	4	1,069	2	
9.01.05	CL	Estrutura Organizacional	100,0	4,637	4	2,911	4	1,270	2	
9.01.06	CL	Alocação dos Profissionais	100,0	3,591	4	1,865	4	2,991	2	
9.01.07	CL	Distribuição de Competências	100,0	3,452	4	3,452	4	0,635	2	
9.01.08	CL	Relações Informais	100,0	4,637	4	12,275	4	0,05	4,579	2
9.01.09	CL	Processos Organizacionais	100,0	1,185	2	3,278	2	1,632	1	
10.01.01	LT	Conhecimento Teórico em ES	100,0	9,445	4	5,487	4	2,657	2	
10.01.02	LT	Inovações Tecnológicas	100,0	6,363	4	3,591	4	0,760	2	
10.01.03	LT	LA da Indústria de Software	100,0	1,185	2	3,278	2	1,632	1	
10.01.04	LT	MP da Indústria de Software	100,0	5,626	2	5,487	2	2,657	1	

Tabela III.5 – Valores críticos de χ^2

gl	$\alpha= 0,1$	$\alpha= 0,05$	$\alpha= 0,01$
1	2,706	3,841	6,635
2	4,605	5,991	9,21
3	6,25	7,815	11,34
4	7,78	9,488	13,28
5	9,24	11,07	15,09
6	10,64	12,59	16,81
7	12,02	14,07	18,48
8	13,36	15,51	20,09
9	14,68	16,92	21,67
10	15,99	18,31	23,21

A tabela III.6 apresenta a média (\bar{x}), a mediana (\tilde{x}), e o desvio padrão (s) da importância atribuída a cada tipo de conhecimento, considerando cada grupo de participantes da pesquisa em separado. Para tal, as ocorrências da categoria *Não Avaliado* nos questionários foram substituídas pela categoria do SPSS v 10.0 que representa não resposta e as demais categorias foram convertidas em valores numéricos, conforme conversão já proposta no questionário (anexo I).

Tabela III.6 – Medidas de Tendência Central e Dispersão

Código	Grupo	Tipo de Conhecimento	Programador			Analista			Gerente			Diretor		
			\bar{x}	\bar{x}	s	\bar{x}	\bar{x}	s	\bar{x}	\bar{x}	s	\bar{x}	\bar{x}	s
4.01	DM	Domínio da Aplicação	2,80	3	1,14	3,13	3	0,86	3,19	3	0,90	3,67	4	0,50
5.01.01	ORG	Missão da Organização	2,54	3	1,00	2,91	3	0,88	2,87	3	1,15	2,78	3	1,20
5.01.02	ORG	Deficiências e Potenciais	2,68	3	1,13	2,84	3	0,9	2,87	3	1,06	2,89	3	1,05
5.01.03	ORG	Diretrizes e Normas	2,74	3	0,97	2,84	3	0,86	2,81	3	1,00	2,78	3	1,30
5.01.04	ORG	Objetivos e Metas	2,77	3	1,11	3,06	3	0,8	2,88	3	1,01	2,89	3	1,05
5.01.05	ORG	Áreas Críticas p/ Objetivos	2,53	3	1,11	2,84	3	0,94	2,90	3	0,91	3,00	3	1,12
5.01.06	ORG	Produtos e/ou Serviços	2,03	2,0	1,09	2,26	2	1,02	2,71	3	1,01	2,56	2	0,73
5.01.07	ORG	Estrutura Organizacional	2,55	2,5	0,95	2,72	3	0,98	2,53	2	1,02	1,88	2	0,83
5.01.08	ORG	Alocação dos Profissionais	2,35	2,0	1,18	2,36	2	1,00	2,22	2	1,13	2,00	2	0,93
5.01.09	ORG	Distribuição de Competências	2,95	3	0,90	2,84	3	1,05	3,16	3	0,85	3,20	3	0,63
5.01.10	ORG	Relações Informais	2,21	2	1,03	1,95	2	1,09	1,94	2	1,13	2,56	3	1,13
5.01.11	ORG	Processos Organizacionais	2,63	3	0,94	2,86	3	0,87	2,91	3	0,69	2,56	3	1,13
5.01.12	ORG	Contratos/Acordos	2,17	2	1,20	2,41	2	1,08	2,42	3	1,18	2,11	2	0,93
6.01.01	OC	Tipos de Software	2,78	3	1,08	2,77	3	1,01	2,81	3	0,93	3,10	3	0,99
6.01.02	OC	Processo de Software	3,31	3	0,69	3,14	3	0,90	3,22	3	0,91	3,30	3,5	0,82
6.01.03	OC	Avaliação do Processo	2,90	3	0,85	3,09	3	0,75	2,75	3	0,92	3,30	4	0,95
6.02	OC	Roteiros de Documentos	2,95	3	1,05	2,94	3	0,87	3,06	3	0,95	2,90	3	1,29
6.03	OC	Itens de Software	3,22	3	0,72	2,99	3	0,81	3,13	3	0,79	3,10	3,5	1,10
6.04.01	MP	Modelagem de Processo	3,08	3	0,94	2,99	3	0,80	3,34	3	0,70	3,20	3	0,79
6.04.02	MP	Gerência de Projeto	2,86	3	1,20	3,09	3	0,84	3,50	4	0,62	3,30	3	0,67
6.04.03	MP	Gerência de Riscos	2,79	3	1,22	2,71	3	0,92	2,97	3	0,97	2,89	3	1,17
6.04.04	MP	Gerência de Qualidade	3,09	4	1,20	3,13	3	0,81	3,47	4	0,67	3,30	3,5	0,95
6.04.05	MP	Gerência de Configuração	2,69	3	1,28	2,56	3	0,87	2,71	3	0,78	3	3	0,71
6.04.06	MP	Documentação	3,40	4	0,84	3,12	3	0,88	3,31	3	0,74	3,30	3,5	0,82
6.04.07	MP	Engenharia de Sistemas	3,12	3	0,95	3,12	3	0,83	3,25	3	0,72	3,20	3	0,92
6.04.08	MP	Análise de Requisitos	3,29	4	0,96	3,40	4	0,74	3,44	4	0,72	3	3	1,05
6.04.09	MP	Projeto (design)	3,30	3,5	0,82	3,21	3	0,78	3,44	4	0,76	3	3	0,67
6.04.10	MP	Codificação	3,46	4	0,78	3,30	3	0,75	3,22	3	0,79	3,6	4	0,70
6.04.11	MP	Testes	3,46	4	0,71	3,43	4	0,69	3,50	4	0,67	3,6	4	0,70
6.04.12	MP	Operação/Suporte a Usuários	2,90	3	0,93	2,94	3	0,91	2,84	3	0,88	3,1	3	0,88
6.04.13	MP	Manutenção	3,28	3	0,82	3,06	3	0,83	3,19	3	0,83	3	3	1,05
6.04.14	MP	Engenharia Reversa e Reeng	2,03	2	1,38	2,17	2	1,14	2,35	2	1,05	1,67	2	1,32
6.04.15	MP	Reuso de Itens de Software	3,07	3	0,88	2,90	3	0,84	2,84	3	0,88	3	3,5	1,33
6.04.16	MP	Melhoria de Processo	3,26	3	0,76	2,96	3	0,70	3,06	3	0,76	3,4	4	0,84
6.04.17	MP	Treinamento	2,90	3	1,12	2,88	3	0,92	2,84	3	0,95	3,2	3,5	1,03
6.04.18	MP	Processo de Aquisição	2,29	2	1,20	2,30	2	0,96	2,23	2	1,06	2,44	3	0,88
6.04.19	MP	Processo de Fornecimento	2,15	2	1,23	2,27	2	0,89	2,31	2	1,09	2,9	3	0,57
6.04.20	MP	Solução de Problemas	3,08	3	0,91	3,16	3	0,75	3,25	4	0,95	3,3	3,5	0,82
6.05.01	LA	Modelagem de Processo	3,00	3	0,97	2,90	3	0,86	3,00	3	0,92	2,89	3	1,27
6.05.02	LA	Gerência de Projeto	2,91	3	1,06	3,08	3	0,79	3,22	3	0,75	3,1	3	0,99
6.05.03	LA	Gerência de Riscos	2,69	3	1,20	2,77	3	1,03	2,77	3	0,92	2,33	2	1,12
6.05.04	LA	Gerência de Qualidade	2,91	3	1,11	3,02	3	0,94	3,06	3	0,95	2,8	3	1,03
6.05.05	LA	Gerência de Configuração	2,61	3	1,32	2,63	3	0,96	2,53	2,5	0,94	2,89	3	0,93
6.05.06	LA	Documentação	3,18	3	0,90	2,93	3	0,98	2,97	3	0,86	3	3	1,05
6.05.07	LA	Engenharia de Sistemas	2,97	3	0,97	3,02	3	0,87	2,97	3	0,90	2,8	3	1,14
6.05.08	LA	Análise de Requisitos	3,03	3	1,07	3,09	3	0,88	3,19	3	0,86	3,3	4	1,06
6.05.09	LA	Projeto (design)	2,90	3	1,01	3,03	3	0,90	3,13	3	0,87	2,8	3	1,23
6.05.10	LA	Codificação	3,20	3	0,75	3,12	3	0,83	2,97	3	0,78	3	4	1,41
6.05.11	LA	Testes	3,17	3	0,86	3,17	3	0,79	3,19	3	0,78	3,3	3	0,67
6.05.12	LA	Operação/Suporte a Usuários	2,97	3	0,99	2,76	3	0,85	2,50	2	0,92	2,7	3	1,06
6.05.13	LA	Manutenção	3,03	3	0,81	3,00	3	0,76	2,77	3	0,92	2,8	3,5	1,48
6.05.14	LA	Engenharia Reversa e Reeng	2,26	3	1,40	2,34	2	1,12	2,30	2	1,15	2	2	1,66
6.05.15	LA	Reuso de Itens de Software	2,93	3	0,98	2,69	3	0,95	2,94	3	0,96	2,6	3	1,35
6.05.16	LA	Melhoria de Processo	2,97	3	0,91	2,84	3	0,98	2,87	3	0,85	2,7	2,5	1,06
6.05.17	LA	Treinamento	2,73	3	1,10	2,69	3	0,95	2,63	3	0,98	2,8	2,5	1,14
6.05.18	LA	Processo de Aquisição	2,20	2	1,13	2,14	2	0,91	2,13	2	1,09	2,67	3	1,58
6.05.19	LA	Processo de Fornecimento	2,13	2	1,31	2,26	2	0,92	2,42	2	0,96	2,9	3	1,29
6.05.20	LA	Solução de Problemas	2,89	3	1,03	2,96	3	0,84	3,09	3	0,89	3,3	4	1,25
6.06.01	DH	Gerência de Projeto	2,65	3	1,07	2,88	3	0,87	3,19	3	0,78	3,4	4	0,97
6.06.02	DH	Gerência de Riscos	2,57	3	1,01	2,55	3	1,03	3,00	3	0,84	2,89	3	1,17
6.06.03	DH	Gerência de Qualidade	2,80	3	1,05	2,92	3	0,85	3,22	3	0,75	3,3	3,5	0,82
6.06.04	DH	Treinamento	2,60	3	1,12	2,75	3	0,92	2,75	3	0,88	3	3	0,94
6.06.05	DH	Métricas de Produto	2,81	3	1,01	2,76	3	0,92	3,09	3	0,93	3,3	4	1,06
6.06.06	DH	Métricas de Processo	2,92	3	0,91	2,74	3	0,95	3,03	3	0,86	3,3	4	1,25
6.07	OC	Respostas para Perguntas	2,77	3	1,06	2,70	3	0,86	3,00	3	0,80	3	3	1,15
7.01.01	FN	Missão da Organização	1,56	2	1,11	1,56	1	1,13	1,35	1	0,98	2,13	2,5	1,36
7.01.02	FN	Deficiências e Potenciais	1,79	2	1,20	1,86	2	1,10	1,77	2	1,27	2,13	2,5	1,36

Tabela III.6 – Medidas de Tendência Central e Dispersão (cont.)

Código	Grupo	Tipo de Conhecimento	Programador			Analista			Gerente			Diretor		
			\bar{x}	\tilde{x}	s	\bar{x}	\tilde{x}	s	\bar{x}	\tilde{x}	s	\bar{x}	\tilde{x}	s
7.01.03	FN	Diretrizes e Normas	1,50	1,5	1,08	1,53	1	1,03	1,42	1	1,03	2	2	1,31
7.01.04	FN	Produtos e/ou Serviços	1,80	2	1,16	1,82	2	1,14	1,96	2	0,94	2,88	3	0,83
7.01.05	FN	Estrutura Organizacional	1,32	1	1,12	1,39	1	0,98	1,12	1	0,91	2	2	1,69
7.01.06	FN	Alocação dos Profissionais	1,12	1	1,11	1,50	2	1,06	1,19	1	0,90	1,88	2	1,73
7.01.07	FN	Distribuição de Competências	1,32	1	1,15	1,75	2	1,18	1,65	1,5	1,02	2,38	2,5	1,41
7.01.08	FN	Relações Informais	1,18	1	1,13	1,25	1	1,07	0,96	1	0,96	1,38	1	1,19
7.01.09	FN	Processos Organizacionais	1,58	2	1,20	1,46	2	0,99	1,12	1	1,07	2	2	0,93
8.01.01	PT	Missão da Organização	2,17	3	1,37	1,72	2	0,96	2,00	2	1,22	2,89	3	0,78
8.01.02	PT	Deficiências e Potenciais	2,23	3	1,38	1,85	2	1	2,13	2	1,26	3,33	3	0,71
8.01.03	PT	Diretrizes e Normas	1,97	2	1,32	1,50	1	0,82	1,96	2	1,2	3,11	3	0,78
8.01.04	PT	Produtos e/ou Serviços	2,00	2	1,39	2,00	2	1,06	2,17	2	1,17	3,11	3	0,78
8.01.05	PT	Estrutura Organizacional	1,93	2	1,23	1,84	2	1,02	1,71	1,5	1,23	2,44	3	1,33
8.01.06	PT	Alocação dos Profissionais	1,93	2	1,31	1,93	2	1,04	1,75	2	1,26	2,78	3	0,97
8.01.07	PT	Distribuição de Competências	1,93	2	1,36	2,04	2	1,04	1,92	2	1,28	3,11	3	0,93
8.01.08	PT	Relações Informais	1,80	2	1,30	1,51	2	1	1,50	1	1,32	2,33	3	1,00
8.01.09	PT	Processos Organizacionais	1,87	2	1,36	1,78	2	0,94	1,79	2	1,35	2,56	3	1,33
9.01.01	CL	Missão da Organização	2,37	2	1,10	2,67	3	1,08	2,72	3	1,16	2,9	3	0,99
9.01.02	CL	Deficiências e Potenciais	2,39	2	0,95	2,72	3	1,02	2,79	3	1,08	2,8	2,5	0,92
9.01.03	CL	Diretrizes e Normas	2,32	2	0,98	2,67	3	1,08	2,76	3	1,12	3	3	0,82
9.01.04	CL	Produtos e/ou Serviços	2,00	2	1,13	2,59	3	1,07	2,69	3	1	3	3	1,05
9.01.05	CL	Estrutura Organizacional	2,32	2	1,22	2,68	3	1,16	2,93	3	1,07	2,7	3	1,42
9.01.06	CL	Alocação dos Profissionais	1,81	2	1,28	2,32	2	1,10	2,55	2	1,12	2,5	3	1,43
9.01.07	CL	Distribuição de Competências	2,10	2	1,40	2,31	2	1,12	2,59	3	1,05	2,5	2,5	1,08
9.01.08	CL	Relações Informais	1,90	2	1,37	2,07	2	1,16	2,45	3	1,15	2,2	2	1,03
9.01.09	CL	Processos Organizacionais	2,50	3	1,14	2,71	3	1,08	2,90	3	1,08	3,1	3	0,74
10.01.01	LT	Conhecimento Teórico em ES	2,86	3	1,05	2,87	3	0,89	3,09	3	0,73	2,90	3	0,99
10.01.02	LT	Inovações Tecnológicas	2,92	3	1	2,90	3	0,86	3,19	3	0,69	2,8	3	1,03
10.01.03	LT	LA da Indústria de Software	2,86	3	0,93	2,88	3	0,81	3,28	3	0,68	3,1	3	0,74
10.01.04	LT	MP da Indústria de Software	2,89	3	1,01	2,84	3	0,80	3,34	3	0,7	2,90	3	0,74

ANEXO IV

Classificação dos Tipos de Conhecimento quanto à Importância

Tabela IV.1 – Tipos de Conhecimento quanto à Importância atribuída por Programadores

Programador (41 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.10	MP	Codificação	1º	7.01.06	FN	Alocação dos Profissionais
1º	6.04.11	MP	Testes	2º	7.01.08	FN	Relações Informais
3º	6.04.06	MP	Documentação	3º	7.01.07	FN	Distribuição de Competências
4º	6.03	OC	Itens de Software	3º	7.01.05	FN	Estrutura Organizacional
4º	6.04.09	MP	Projeto (design)	5º	7.01.03	FN	Diretrizes e Normas
6º	6.04.13	MP	Manutenção	6º	7.01.09	FN	Processos Organizacionais
6º	6.05.10	LA	Codificação	7º	7.01.01	FN	Missão da Organização
8º	6.05.11	LA	Testes	8º	8.01.08	PT	Relações Informais
9º	6.01.02	OC	Processo de Software	9º	9.01.08	CL	Relações Informais
10º	6.04.08	MP	Análise de Requisitos	10º	9.01.06	CL	Alocação dos Profissionais
11º	6.05.06	LA	Documentação	10º	8.01.09	PT	Processos Organizacionais
11º	6.05.08	LA	Análise de Requisitos	12º	8.01.03	PT	Diretrizes e Normas
13º	6.05.15	LA	Reuso de Itens de Software	13º	8.01.07	PT	Distribuição de Competências
14º	6.04.01	MP	Modelagem de Processo	13º	8.01.06	PT	Alocação dos Profissionais
15º	6.05.12	LA	Operação/Suporte a Usuários	13º	8.01.05	PT	Estrutura Organizacional
16º	6.02	OC	Roteiros de Documentos	13º	8.01.04	PT	Produtos e/ou Serviços
17º	6.01.03	OC	Avaliação do Processo	17º	7.01.02	FN	Deficiências e Potenciais
17º	6.05.16	LA	Melhoria de Processo	18º	9.01.04	CL	Produtos e/ou Serviços
19º	5.01.09	ORG	Distribuição de Competências	19º	9.01.07	CL	Distribuição de Competências
20º	6.04.04	MP	Gerência de Qualidade	19º	7.01.04	FN	Produtos e/ou Serviços
21º	6.04.07	MP	Engenharia de Sistemas	21º	8.01.01	PT	Missão da Organização
22º	0.01.02	LT	Inovações Tecnológicas	22º	8.01.02	PT	Deficiências e Potenciais
23º	0.01.04	LT	LA da Indústria de Software	23º	9.01.05	CL	Estrutura Organizacional
24º	0.01.03	LT	MP da Indústria de Software	24º	9.01.09	CL	Processos Organizacionais
25º	6.04.05	MP	Gerência de Configuração	24º	5.01.06	ORG	Produtos e/ou Serviços

Tabela IV.2 – Tipos de Conhecimento quanto à Importância atribuída por Programadores com Pouca Experiência

Programador com Pouca Experiência (14 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.10	MP	Codificação	1º	7.01.06	FN	Alocação dos Profissionais
1º	6.05.08	LA	Análise de Requisitos	2º	7.01.08	FN	Relações Informais
1º	6.05.11	LA	Testes	3º	9.01.04	CL	Produtos e/ou Serviços
4º	6.04.11	MP	Testes	3º	9.01.01	CL	Missão da Organização
5º	6.05.10	LA	Codificação	3º	7.01.01	FN	Missão da Organização
6º	6.05.15	LA	Reuso de Itens de Software	6º	7.01.05	FN	Estrutura Organizacional
7º	6.04.06	MP	Documentação	7º	9.01.03	CL	Diretrizes e Normas
7º	6.04.15	MP	Reuso de Itens de Software	7º	7.01.09	FN	Processos Organizacionais
7º	0.01.04	LT	LA da Indústria de Software	7º	7.01.07	FN	Distribuição de Competências
10º	0.01.02	LT	Inovações Tecnológicas	10º	9.01.05	CL	Estrutura Organizacional
11º	6.04.09	MP	Projeto (design)	11º	7.01.03	FN	Diretrizes e Normas
11º	6.05.06	LA	Documentação	11º	7.01.02	FN	Deficiências e Potenciais
11º	0.01.03	LT	MP da Indústria de Software	13º	5.01.12	ORG	Contratos/Acordos
14º	6.04.13	MP	Manutenção	14º	9.01.02	CL	Deficiências e Potenciais
14º	0.01.01	LT	Conhecimento Teórico em ES	14º	7.01.04	FN	Produtos e/ou Serviços
16º	6.06.03	DH	Gerência de Qualidade	16º	8.01.08	PT	Relações Informais
17º	6.05.02	LA	Gerência de Projeto	16º	6.04.14	MP	Engenharia Reversa e Reeng
17º	6.06.05	DH	Métricas de Produto	18º	8.01.09	PT	Processos Organizacionais
19º	6.04.04	MP	Gerência de Qualidade	18º	8.01.05	PT	Estrutura Organizacional
19º	6.04.17	MP	Treinamento	20º	6.05.14	LA	Engenharia Reversa e Reeng
21º	6.04.20	MP	Solução de Problemas	21º	8.01.04	PT	Produtos e/ou Serviços
22º	6.04.08	MP	Análise de Requisitos	21º	8.01.03	PT	Diretrizes e Normas
23º	6.06.01	DH	Gerência de Projeto	23º	8.01.07	PT	Distribuição de Competências
24º	6.05.01	LA	Modelagem de Processo	23º	8.01.01	PT	Missão da Organização
25º	6.04.05	MP	Gerência de Configuração	25º	8.01.02	PT	Deficiências e Potenciais

Tabela IV.3 – Tipos de Conhecimento quanto à Importância atribuída por Programadores com Média Experiência

Programador com Média Experiência (18 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.11	MP	Testes	1º	7.01.08	FN	Relações Informais
2º	6.04.10	MP	Codificação	1º	7.01.07	FN	Distribuição de Competências
3º	6.04.06	MP	Documentação	1º	7.01.06	FN	Alocação dos Profissionais
3º	6.04.08	MP	Análise de Requisitos	4º	7.01.05	FN	Estrutura Organizacional
3º	6.04.09	MP	Projeto (design)	5º	9.01.06	CL	Alocação dos Profissionais
3º	6.04.13	MP	Manutenção	5º	8.01.08	PT	Relações Informais
7º	6.03	OC	Itens de Software	7º	9.01.07	CL	Distribuição de Competências
8º	6.01.02	OC	Processo de Software	7º	8.01.07	PT	Distribuição de Competências
9º	6.04.15	MP	Reuso de Itens de Software	7º	7.01.03	FN	Diretrizes e Normas
9º	6.05.10	LA	Codificação	10º	8.01.09	PT	Processos Organizacionais
11º	6.04.16	MP	Melhoria de Processo	10º	8.01.06	PT	Alocação dos Profissionais
12º	6.05.11	LA	Testes	10º	8.01.05	PT	Estrutura Organizacional
13º	6.04.07	MP	Engenharia de Sistemas	10º	8.01.04	PT	Produtos e/ou Serviços
13º	6.04.17	MP	Treinamento	14º	7.01.09	FN	Processos Organizacionais
13º	6.05.06	LA	Documentação	15º	9.01.08	CL	Relações Informais
13º	6.05.09	LA	Projeto (design)	15º	6.05.19	LA	Processo de Fornecimento
17º	5.01.04	ORG	Objetivos e Metas	17º	8.01.01	PT	Missão da Organização
18º	6.02	OC	Roteiros de Documentos	18º	9.01.04	CL	Produtos e/ou Serviços
18º	6.05.08	LA	Análise de Requisitos	19º	7.01.01	FN	Missão da Organização
18º	6.07	OC	Respostas para Perguntas	20º	7.01.02	FN	Deficiências e Potenciais
21º	6.04.04	MP	Gerência de Qualidade	21º	5.01.06	ORG	Produtos e/ou Serviços
22º	6.05.07	LA	Engenharia de Sistemas	22º	7.01.04	FN	Produtos e/ou Serviços
23º	6.04.05	MP	Gerência de Configuração	22º	6.04.19	MP	Processo de Fornecimento
24º	6.05.01	LA	Modelagem de Processo	22º	6.04.14	MP	Engenharia Reversa e Reeng
24º	6.05.04	LA	Gerência de Qualidade	25º	5.01.10	ORG	Relações Informais

Tabela IV.4 – Tipos de Conhecimento quanto à Importância atribuída por Programadores com Muita Experiência

Programador com Muita Experiência (9 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.10	MP	Codificação	1º	7.01.06	FN	Alocação dos Profissionais
1º	6.04.11	MP	Testes	2º	8.01.06	PT	Alocação dos Profissionais
3º	6.04.06	MP	Documentação	2º	7.01.08	FN	Relações Informais
4º	5.01.09	ORG	Distribuição de Competências	4º	8.01.07	PT	Distribuição de Competências
4º	6.04.01	MP	Modelagem de Processo	4º	8.01.04	PT	Produtos e/ou Serviços
4º	6.04.20	MP	Solução de Problemas	4º	8.01.03	PT	Diretrizes e Normas
7º	6.03	OC	Itens de Software	4º	7.01.07	FN	Distribuição de Competências
7º	6.04.09	MP	Projeto (design)	8º	8.01.09	PT	Processos Organizacionais
7º	6.04.13	MP	Manutenção	8º	7.01.05	FN	Estrutura Organizacional
7º	6.05.06	LA	Documentação	8º	7.01.04	FN	Produtos e/ou Serviços
11º	6.04.08	MP	Análise de Requisitos	8º	7.01.03	FN	Diretrizes e Normas
11º	6.04.12	MP	Operação/Suporte a Usuários	8º	7.01.02	FN	Deficiências e Potenciais
13º	4.01	DM	Domínio da Aplicação	13º	8.01.08	PT	Relações Informais
13º	5.01.01	ORG	Missão da Organização	13º	8.01.05	PT	Estrutura Organizacional
13º	6.02	OC	Roteiros de Documentos	13º	7.01.09	FN	Processos Organizacionais
16º	5.01.02	ORG	Deficiências e Potenciais	13º	7.01.01	FN	Missão da Organização
16º	6.04.16	MP	Melhoria de Processo	17º	8.01.01	PT	Missão da Organização
16º	6.05.09	LA	Projeto (design)	18º	8.01.02	PT	Deficiências e Potenciais
19º	5.01.04	ORG	Objetivos e Metas	18º	6.04.14	MP	Engenharia Reversa e Reeng
19º	6.04.04	MP	Gerência de Qualidade	20º	9.01.08	CL	Relações Informais
19º	6.04.07	MP	Engenharia de Sistemas	20º	6.05.18	LA	Processo de Aquisição
19º	6.04.15	MP	Reuso de Itens de Software	22º	9.01.06	CL	Alocação dos Profissionais
19º	6.05.08	LA	Análise de Requisitos	23º	6.05.19	LA	Processo de Fornecimento
24º	5.01.05	ORG	Áreas Críticas para Objetivos	24º	5.01.12	ORG	Contratos/Acordos
24º	6.05.01	LA	Modelagem de Processo	24º	5.01.06	ORG	Produtos e/ou Serviços

Tabela IV.5 – Tipos de Conhecimento quanto à Importância atribuída por Programadores de Micro Empresas

Programador de Micro Empresa (2 participantes)							
Mais Importantes				Menos Importantes			
1º	5.01.09	ORG	Distribuição de Competências	1º	8.01.01	PT	Missão da Organização
1º	5.01.10	ORG	Relações Informais	1º	7.01.08	FN	Relações Informais
1º	0.01.01	LT	Conhecimento Teórico em ES	2º	8.01.05	PT	Estrutura Organizacional
1º	0.01.02	LT	Inovações Tecnológicas	2º	8.01.04	PT	Produtos e/ou Serviços
1º	0.01.03	LT	MP da Indústria de Software	2º	7.01.09	FN	Processos Organizacionais
1º	0.01.04	LT	LA da Indústria de Software	2º	7.01.07	FN	Distribuição de Competências
7º	5.01.02	ORG	Deficiências e Potenciais	2º	7.01.06	FN	Alocação dos Profissionais
7º	5.01.03	ORG	Diretrizes e Normas	2º	7.01.05	FN	Estrutura Organizacional
7º	5.01.04	ORG	Objetivos e Metas	2º	7.01.04	FN	Produtos e/ou Serviços
7º	6.01.02	OC	Processo de Software	2º	7.01.01	FN	Missão da Organização
7º	6.01.03	OC	Avaliação do Processo	2º	6.06.04	DH	Treinamento
7º	6.03	OC	Itens de Software	2º	6.05.17	LA	Treinamento
7º	6.04.03	MP	Gerência de Riscos	2º	6.05.14	LA	Engenharia Reversa e Reeng
7º	6.04.08	MP	Análise de Requisitos	2º	6.05.13	LA	Manutenção
7º	6.04.10	MP	Codificação	2º	6.05.05	LA	Gerência de Configuração
7º	6.04.11	MP	Testes	2º	6.05.04	LA	Gerência de Qualidade
7º	6.04.16	MP	Melhoria de Processo	2º	6.04.17	MP	Treinamento
7º	6.04.20	MP	Solução de Problemas	2º	6.04.14	MP	Engenharia Reversa e Reeng
7º	6.05.03	LA	Gerência de Riscos	2º	6.04.06	MP	Documentação
7º	6.05.06	LA	Documentação	2º	6.04.02	MP	Gerência de Projeto
7º	6.05.08	LA	Análise de Requisitos	2º	6.02	OC	Roteiros de Documentos
7º	6.05.11	LA	Testes	2º	5.01.08	ORG	Alocação dos Profissionais
7º	6.05.16	LA	Melhoria de Processo	2º	5.01.07	ORG	Estrutura Organizacional
7º	6.05.20	LA	Solução de Problemas	2º	5.01.06	ORG	Produtos e/ou Serviços
7º	7.01.02	FN	Deficiências e Potenciais	2º	5.01.05	ORG	Áreas Críticas para Objetivos

Tabela IV.6 – Tipos de Conhecimento quanto à Importância atribuída por Programadores de Pequenas Empresas

Programador de Pequena Empresa (7 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.06	MP	Documentação	1º	8.01.07	PT	Distribuição de Competências
1º	6.04.08	MP	Análise de Requisitos	1º	8.01.03	PT	Diretrizes e Normas
1º	6.04.09	MP	Projeto (design)	1º	7.01.08	FN	Relações Informais
1º	6.04.10	MP	Codificação	1º	7.01.06	FN	Alocação dos Profissionais
5º	6.03	OC	Itens de Software	5º	9.01.06	CL	Alocação dos Profissionais
5º	6.04.11	MP	Testes	5º	8.01.05	PT	Estrutura Organizacional
5º	6.04.13	MP	Manutenção	5º	8.01.04	PT	Produtos e/ou Serviços
8º	5.01.09	ORG	Distribuição de Competências	8º	8.01.09	PT	Processos Organizacionais
8º	6.04.15	MP	Reuso de Itens de Software	9º	7.01.07	FN	Distribuição de Competências
8º	6.05.10	LA	Codificação	9º	7.01.05	FN	Estrutura Organizacional
11º	5.01.04	ORG	Objetivos e Metas	9º	7.01.03	FN	Diretrizes e Normas
11º	6.04.01	MP	Modelagem de Processo	12º	9.01.08	CL	Relações Informais
11º	6.04.02	MP	Gerência de Projeto	12º	9.01.05	CL	Estrutura Organizacional
11º	6.04.04	MP	Gerência de Qualidade	12º	7.01.09	FN	Processos Organizacionais
11º	6.04.20	MP	Solução de Problemas	12º	7.01.04	FN	Produtos e/ou Serviços
11º	6.05.09	LA	Projeto (design)	12º	7.01.01	FN	Missão da Organização
17º	6.01.02	OC	Processo de Software	17º	9.01.04	CL	Produtos e/ou Serviços
17º	6.04.12	MP	Operação/Suporte a Usuários	18º	9.01.07	CL	Distribuição de Competências
17º	6.04.16	MP	Melhoria de Processo	18º	9.01.03	CL	Diretrizes e Normas
20º	4.01	DM	Domínio da Aplicação	18º	5.01.12	ORG	Contratos/Acordos
20º	5.01.03	ORG	Diretrizes e Normas	18º	5.01.10	ORG	Relações Informais
20º	6.04.07	MP	Engenharia de Sistemas	22º	5.01.06	ORG	Produtos e/ou Serviços
20º	6.05.08	LA	Análise de Requisitos	23º	6.05.19	LA	Processo de Fornecimento
24º	6.02	OC	Roteiros de Documentos	23º	6.05.18	LA	Processo de Aquisição
24º	6.04.17	MP	Treinamento	25º	6.06.04	DH	Treinamento

Tabela IV.7 – Tipos de Conhecimento quanto à Importância atribuída por Programadores de Empresas Médias

Programador de Média Empresa (14 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.11	MP	Testes	1º	7.01.08	FN	Relações Informais
2º	6.04.10	MP	Codificação	2º	7.01.06	FN	Alocação dos Profissionais
3º	6.01.02	OC	Processo de Software	3º	7.01.09	FN	Processos Organizacionais
4º	6.03	OC	Itens de Software	3º	7.01.05	FN	Estrutura Organizacional
4º	6.04.06	MP	Documentação	5º	7.01.03	FN	Diretrizes e Normas
4º	6.04.13	MP	Manutenção	6º	7.01.07	FN	Distribuição de Competências
4º	6.05.11	LA	Testes	6º	7.01.02	FN	Deficiências e Potenciais
8º	6.04.09	MP	Projeto (design)	8º	8.01.08	PT	Relações Informais
9º	6.02	OC	Roteiros de Documentos	8º	8.01.06	PT	Alocação dos Profissionais
9º	6.04.20	MP	Solução de Problemas	8º	6.05.19	LA	Processo de Fornecimento
9º	6.05.10	LA	Codificação	11º	8.01.07	PT	Distribuição de Competências
12º	6.05.12	LA	Operação/Suporte a Usuários	11º	6.05.18	LA	Processo de Aquisição
13º	6.04.01	MP	Modelagem de Processo	13º	8.01.09	PT	Processos Organizacionais
13º	6.04.16	MP	Melhoria de Processo	13º	8.01.05	PT	Estrutura Organizacional
15º	5.01.01	ORG	Missão da Organização	13º	8.01.04	PT	Produtos e/ou Serviços
15º	5.01.05	ORG	Áreas Críticas para Objetivos	13º	8.01.03	PT	Diretrizes e Normas
15º	6.04.12	MP	Operação/Suporte a Usuários	13º	7.01.04	FN	Produtos e/ou Serviços
15º	6.05.06	LA	Documentação	13º	6.04.18	MP	Processo de Aquisição
19º	5.01.02	ORG	Deficiências e Potenciais	19º	8.01.02	PT	Deficiências e Potenciais
19º	5.01.04	ORG	Objetivos e Metas	19º	8.01.01	PT	Missão da Organização
19º	6.04.08	MP	Análise de Requisitos	21º	6.04.14	MP	Engenharia Reversa e Reeng
19º	6.05.09	LA	Projeto (design)	21º	6.04.05	MP	Gerência de Configuração
23º	6.04.17	MP	Treinamento	23º	9.01.06	CL	Alocação dos Profissionais
24º	4.01	DM	Domínio da Aplicação	23º	6.04.03	MP	Gerência de Riscos
24º	6.05.08	LA	Análise de Requisitos	25º	6.04.02	MP	Gerência de Projeto

Tabela IV.8 – Tipos de Conhecimento quanto à Importância atribuída por Programadores de Grandes Empresas

Programador de Grande Empresa (18 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.10	MP	Codificação	1º	6.01.03	OC	Avaliação do Processo
1º	6.04.11	MP	Testes	2º	5.01.10	ORG	Relações Informais
3º	6.04.06	MP	Documentação	3º	7.01.04	FN	Produtos e/ou Serviços
4º	6.04.15	MP	Reuso de Itens de Software	4º	6.04.14	MP	Engenharia Reversa e Reeng
4º	6.05.10	LA	Codificação	5º	5.01.06	ORG	Produtos e/ou Serviços
6º	6.04.09	MP	Projeto (design)	5º	6.05.19	LA	Processo de Fornecimento
6º	6.05.08	LA	Análise de Requisitos	5º	7.01.02	FN	Deficiências e Potenciais
8º	6.04.13	MP	Manutenção	5º	8.01.05	PT	Estrutura Organizacional
9º	6.01.02	OC	Processo de Software	5º	8.01.07	PT	Distribuição de Competências
9º	6.04.16	MP	Melhoria de Processo	10º	8.01.06	PT	Alocação dos Profissionais
9º	6.06.03	DH	Gerência de Qualidade	11º	8.01.08	PT	Relações Informais
12º	6.04.08	MP	Análise de Requisitos	12º	7.01.01	FN	Missão da Organização
12º	6.05.06	LA	Documentação	12º	7.01.09	FN	Processos Organizacionais
12º	0.01.03	LT	MP da Indústria de Software	12º	8.01.09	PT	Processos Organizacionais
12º	0.01.04	LT	LA da Indústria de Software	15º	7.01.03	FN	Diretrizes e Normas
16º	0.01.01	LT	Conhecimento Teórico em ES	16º	7.01.05	FN	Estrutura Organizacional
16º	0.01.02	LT	Inovações Tecnológicas	16º	7.01.08	FN	Relações Informais
18º	6.04.04	MP	Gerência de Qualidade	18º	9.01.01	CL	Missão da Organização
18º	6.06.05	DH	Métricas de Produto	18º	9.01.02	CL	Deficiências e Potenciais
20º	6.04.17	MP	Treinamento	18º	9.01.03	CL	Diretrizes e Normas
21º	6.02	OC	Roteiros de Documentos	21º	7.01.07	FN	Distribuição de Competências
21º	6.05.16	LA	Melhoria de Processo	22º	9.01.06	CL	Alocação dos Profissionais
23º	6.04.05	MP	Gerência de Configuração	22º	9.01.08	CL	Relações Informais
23º	6.04.07	MP	Engenharia de Sistemas	24º	7.01.06	FN	Alocação dos Profissionais
23º	6.04.20	MP	Solução de Problemas	25º	9.01.07	CL	Distribuição de Competências

Tabela IV.9 – Tipos de Conhecimento quanto à Importância atribuída por Programadores considerando Projetos Pequenos

Programador considerando Projeto Pequeno (7 participantes)							
Mais Importantes				Menos Importantes			
1º	6.01.02	OC	Processo de Software	1º	7.01.08	FN	Relações Informais
1º	6.04.11	MP	Testes	2º	7.01.06	FN	Alocação dos Profissionais
3º	6.04.10	MP	Codificação	3º	7.01.09	FN	Processos Organizacionais
3º	6.05.11	LA	Testes	3º	7.01.05	FN	Estrutura Organizacional
5º	6.05.08	LA	Análise de Requisitos	3º	7.01.01	FN	Missão da Organização
5º	6.05.16	LA	Melhoria de Processo	6º	7.01.03	FN	Diretrizes e Normas
7º	6.01.03	OC	Avaliação do Processo	6º	6.06.04	DH	Treinamento
7º	6.03	OC	Itens de Software	6º	6.06.02	DH	Gerência de Riscos
7º	6.04.06	MP	Documentação	6º	5.01.06	ORG	Produtos e/ou Serviços
7º	6.04.09	MP	Projeto (design)	10º	8.01.08	PT	Relações Informais
7º	6.04.13	MP	Manutenção	10º	8.01.01	PT	Missão da Organização
7º	6.04.16	MP	Melhoria de Processo	10º	7.01.02	FN	Deficiências e Potenciais
7º	6.05.10	LA	Codificação	10º	6.04.14	MP	Engenharia Reversa e Reeng
7º	6.05.12	LA	Operação/Suporte a Usuários	10º	5.01.12	ORG	Contratos/Acordos
15º	4.01	DM	Domínio da Aplicação	10º	5.01.07	ORG	Estrutura Organizacional
15º	5.01.10	ORG	Relações Informais	16º	8.01.03	PT	Diretrizes e Normas
15º	6.04.08	MP	Análise de Requisitos	16º	7.01.07	FN	Distribuição de Competências
15º	6.04.15	MP	Reuso de Itens de Software	18º	7.01.04	FN	Produtos e/ou Serviços
15º	6.04.17	MP	Treinamento	18º	6.05.14	LA	Engenharia Reversa e Reeng
20º	6.04.01	MP	Modelagem de Processo	18º	6.05.05	LA	Gerência de Configuração
21º	5.01.04	ORG	Objetivos e Metas	21º	6.04.18	MP	Processo de Aquisição
22º	5.01.03	ORG	Diretrizes e Normas	21º	5.01.05	ORG	Áreas Críticas para Objetivos
22º	6.04.07	MP	Engenharia de Sistemas	21º	5.01.02	ORG	Deficiências e Potenciais
24º	6.04.04	MP	Gerência de Qualidade	21º	5.01.01	ORG	Missão da Organização
25º	6.04.05	MP	Gerência de Configuração	25º	0.01.01	LT	Conhecimento Teórico em ES

Tabela IV.10 – Tipos de Conhecimento quanto à Importância atribuída por Programadores considerando Projetos Médios

Programador considerando Projeto Médio (13 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.10	MP	Codificação	1º	7.01.06	FN	Alocação dos Profissionais
1º	6.04.11	MP	Testes	2º	9.01.08	CL	Relações Informais
1º	6.04.13	MP	Manutenção	2º	7.01.07	FN	Distribuição de Competências
4º	6.04.06	MP	Documentação	4º	7.01.08	FN	Relações Informais
5º	6.04.09	MP	Projeto (design)	5º	9.01.04	CL	Produtos e/ou Serviços
6º	6.04.12	MP	Operação/Suporte a Usuários	6º	7.01.05	FN	Estrutura Organizacional
6º	6.05.06	LA	Documentação	7º	7.01.09	FN	Processos Organizacionais
8º	6.03	OC	Itens de Software	7º	7.01.03	FN	Diretrizes e Normas
8º	0.01.01	LT	Conhecimento Teórico em ES	9º	7.01.01	FN	Missão da Organização
10º	0.01.02	LT	Inovações Tecnológicas	10º	8.01.09	PT	Processos Organizacionais
11º	6.01.01	OC	Tipos de Software	11º	6.05.19	LA	Processo de Fornecimento
12º	6.01.02	OC	Processo de Software	12º	8.01.07	PT	Distribuição de Competências
12º	6.04.01	MP	Modelagem de Processo	12º	7.01.02	FN	Deficiências e Potenciais
14º	0.01.04	LT	LA da Indústria de Software	12º	6.04.14	MP	Engenharia Reversa e Reeng
15º	5.01.09	ORG	Distribuição de Competências	15º	7.01.04	FN	Produtos e/ou Serviços
15º	6.02	OC	Roteiros de Documentos	15º	6.05.18	LA	Processo de Aquisição
15º	6.06.04	DH	Treinamento	15º	5.01.12	ORG	Contratos/Acordos
15º	0.01.03	LT	MP da Indústria de Software	18º	6.04.19	MP	Processo de Fornecimento
19º	6.04.02	MP	Gerência de Projeto	19º	6.05.14	LA	Engenharia Reversa e Reeng
20º	5.01.07	ORG	Estrutura Organizacional	19º	6.04.18	MP	Processo de Aquisição
20º	6.05.01	LA	Modelagem de Processo	21º	5.01.10	ORG	Relações Informais
22º	4.01	DM	Domínio da Aplicação	22º	5.01.06	ORG	Produtos e/ou Serviços
23º	6.04.05	MP	Gerência de Configuração	23º	5.01.04	ORG	Objetivos e Metas
24º	5.01.11	ORG	Processos Organizacionais	24º	5.01.03	ORG	Diretrizes e Normas
25º	6.04.07	MP	Engenharia de Sistemas	25º	5.01.01	ORG	Missão da Organização

Tabela IV.11 – Tipos de Conhecimento quanto à Importância atribuída por Programadores considerando Projetos Grandes

Programador considerando Projeto Grande (10 participantes)							
Mais Importantes				Menos Importantes			
1 ^o	6.04.10	MP	Codificação	1 ^o	8.01.04	PT	Produtos e/ou Serviços
2 ^o	6.04.06	MP	Documentação	2 ^o	8.01.05	PT	Estrutura Organizacional
3 ^o	6.01.02	OC	Processo de Software	3 ^o	8.01.06	PT	Alocação dos Profissionais
3 ^o	6.04.11	MP	Testes	3 ^o	7.01.08	FN	Relações Informais
3 ^o	6.05.10	LA	Codificação	3 ^o	7.01.06	FN	Alocação dos Profissionais
6 ^o	6.02	OC	Roteiros de Documentos	6 ^o	8.01.08	PT	Relações Informais
6 ^o	6.03	OC	Itens de Software	6 ^o	8.01.07	PT	Distribuição de Competências
6 ^o	6.04.04	MP	Gerência de Qualidade	6 ^o	8.01.03	PT	Diretrizes e Normas
6 ^o	6.04.16	MP	Melhoria de Processo	6 ^o	7.01.07	FN	Distribuição de Competências
6 ^o	6.06.06	DH	Métricas de Processo	6 ^o	7.01.05	FN	Estrutura Organizacional
11 ^o	6.04.09	MP	Projeto (design)	11 ^o	8.01.09	PT	Processos Organizacionais
11 ^o	6.04.15	MP	Reuso de Itens de Software	11 ^o	7.01.04	FN	Produtos e/ou Serviços
11 ^o	6.06.05	DH	Métricas de Produto	13 ^o	9.01.05	CL	Estrutura Organizacional
14 ^o	6.04.01	MP	Modelagem de Processo	13 ^o	8.01.02	PT	Deficiências e Potenciais
14 ^o	6.05.11	LA	Testes	13 ^o	7.01.03	FN	Diretrizes e Normas
16 ^o	6.07	OC	Respostas para Perguntas	13 ^o	6.04.19	MP	Processo de Fornecimento
17 ^o	4.01	DM	Domínio da Aplicação	17 ^o	7.01.09	FN	Processos Organizacionais
17 ^o	5.01.04	ORG	Objetivos e Metas	17 ^o	7.01.01	FN	Missão da Organização
17 ^o	6.05.15	LA	Reuso de Itens de Software	17 ^o	6.05.19	LA	Processo de Fornecimento
20 ^o	5.01.03	ORG	Diretrizes e Normas	17 ^o	5.01.06	ORG	Produtos e/ou Serviços
20 ^o	5.01.11	ORG	Processos Organizacionais	21 ^o	8.01.01	PT	Missão da Organização
20 ^o	6.05.04	LA	Gerência de Qualidade	21 ^o	7.01.02	FN	Deficiências e Potenciais
23 ^o	6.04.20	MP	Solução de Problemas	21 ^o	5.01.08	ORG	Alocação dos Profissionais
24 ^o	6.04.07	MP	Engenharia de Sistemas	24 ^o	6.05.18	LA	Processo de Aquisição
25 ^o	6.05.17	LA	Treinamento	25 ^o	6.04.17	MP	Treinamento

Tabela IV.12 – Tipos de Conhecimento quanto à Importância atribuída por Programadores considerando Projetos de Manutenção

Programador considerando Projeto Manutenção (11 participantes)							
Mais Importantes				Menos Importantes			
1 ^o	6.03	OC	Itens de Software	1 ^o	7.01.05	FN	Estrutura Organizacional
1 ^o	6.04.11	MP	Testes	2 ^o	7.01.06	FN	Alocação dos Profissionais
3 ^o	6.04.09	MP	Projeto (design)	3 ^o	8.01.08	PT	Relações Informais
3 ^o	6.04.10	MP	Codificação	3 ^o	8.01.07	PT	Distribuição de Competências
3 ^o	6.04.13	MP	Manutenção	3 ^o	8.01.06	PT	Alocação dos Profissionais
6 ^o	6.04.06	MP	Documentação	3 ^o	8.01.03	PT	Diretrizes e Normas
6 ^o	6.04.08	MP	Análise de Requisitos	3 ^o	7.01.03	FN	Diretrizes e Normas
6 ^o	6.05.10	LA	Codificação	3 ^o	7.01.01	FN	Missão da Organização
6 ^o	6.05.11	LA	Testes	9 ^o	8.01.09	PT	Processos Organizacionais
10 ^o	6.01.02	OC	Processo de Software	9 ^o	8.01.05	PT	Estrutura Organizacional
10 ^o	6.04.20	MP	Solução de Problemas	9 ^o	8.01.04	PT	Produtos e/ou Serviços
10 ^o	6.05.12	LA	Operação/Suporte a Usuários	9 ^o	7.01.09	FN	Processos Organizacionais
13 ^o	6.01.01	OC	Tipos de Software	9 ^o	7.01.08	FN	Relações Informais
13 ^o	6.04.15	MP	Reuso de Itens de Software	9 ^o	7.01.07	FN	Distribuição de Competências
13 ^o	6.05.09	LA	Projeto (design)	15 ^o	8.01.02	PT	Deficiências e Potenciais
13 ^o	6.05.15	LA	Reuso de Itens de Software	15 ^o	8.01.01	PT	Missão da Organização
17 ^o	5.01.04	ORG	Objetivos e Metas	15 ^o	7.01.02	FN	Deficiências e Potenciais
17 ^o	6.01.03	OC	Avaliação do Processo	18 ^o	7.01.04	FN	Produtos e/ou Serviços
19 ^o	5.01.01	ORG	Missão da Organização	19 ^o	9.01.08	CL	Relações Informais
19 ^o	5.01.02	ORG	Deficiências e Potenciais	19 ^o	9.01.06	CL	Alocação dos Profissionais
21 ^o	4.01	DM	Domínio da Aplicação	21 ^o	9.01.07	CL	Distribuição de Competências
21 ^o	5.01.05	ORG	Áreas Críticas para Objetivos	21 ^o	6.06.05	DH	Métricas de Produto
21 ^o	6.02	OC	Roteiros de Documentos	23 ^o	5.01.10	ORG	Relações Informais
24 ^o	6.04.02	MP	Gerência de Projeto	24 ^o	5.01.12	ORG	Contratos/Acordos
25 ^o	6.04.14	MP	Engenharia Reversa e Reeng	25 ^o	5.01.06	ORG	Produtos e/ou Serviços

Tabela IV.13 – Tipos de Conhecimento quanto à Importância atribuída por Analistas

Analista (70 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.11	MP	Testes	1º	7.01.08	FN	Relações Informais
2º	6.04.08	MP	Análise de Requisitos	2º	7.01.05	FN	Estrutura Organizacional
2º	6.04.10	MP	Codificação	3º	8.01.03	PT	Diretrizes e Normas
4º	6.05.11	LA	Testes	4º	7.01.09	FN	Processos Organizacionais
5º	6.04.09	MP	Projeto (design)	5º	8.01.08	PT	Relações Informais
6º	6.01.02	OC	Processo de Software	6º	7.01.06	FN	Alocação dos Profissionais
7º	4.01	DM	Domínio da Aplicação	6º	7.01.03	FN	Diretrizes e Normas
8º	6.04.06	MP	Documentação	8º	7.01.01	FN	Missão da Organização
8º	6.04.20	MP	Solução de Problemas	9º	8.01.01	PT	Missão da Organização
8º	6.05.10	LA	Codificação	10º	8.01.09	PT	Processos Organizacionais
11º	5.01.04	ORG	Objetivos e Metas	10º	7.01.07	FN	Distribuição de Competências
11º	6.04.13	MP	Manutenção	12º	8.01.02	PT	Deficiências e Potenciais
13º	6.04.02	MP	Gerência de Projeto	12º	7.01.04	FN	Produtos e/ou Serviços
14º	6.01.03	OC	Avaliação do Processo	14º	8.01.05	PT	Estrutura Organizacional
14º	6.04.04	MP	Gerência de Qualidade	15º	7.01.02	FN	Deficiências e Potenciais
14º	6.05.08	LA	Análise de Requisitos	16º	8.01.06	PT	Alocação dos Profissionais
17º	6.04.07	MP	Engenharia de Sistemas	16º	8.01.04	PT	Produtos e/ou Serviços
18º	5.01.01	ORG	Missão da Organização	18º	9.01.08	CL	Relações Informais
19º	6.02	OC	Roteiros de Documentos	19º	6.04.14	MP	Engenharia Reversa e Reeng
19º	6.04.12	MP	Operação/Suporte a Usuários	20º	9.01.07	CL	Distribuição de Competências
19º	6.05.09	LA	Projeto (design)	21º	9.01.06	CL	Alocação dos Profissionais
22º	6.05.06	LA	Documentação	22º	9.01.04	CL	Produtos e/ou Serviços
23º	6.05.04	LA	Gerência de Qualidade	23º	9.01.05	CL	Estrutura Organizacional
23º	6.05.07	LA	Engenharia de Sistemas	24º	6.06.06	DH	Métricas de Processo
25º	6.05.16	LA	Melhoria de Processo	25º	6.06.04	DH	Treinamento

Tabela IV.14 – Tipos de Conhecimento quanto à Importância atribuída por Analistas com Pouca Experiência

Analista com Pouca Experiência (30 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.11	MP	Testes	1º	7.01.08	FN	Relações Informais
2º	6.04.10	MP	Codificação	2º	8.01.08	PT	Relações Informais
3º	6.04.08	MP	Análise de Requisitos	2º	8.01.03	PT	Diretrizes e Normas
4º	6.05.11	LA	Testes	4º	7.01.05	FN	Estrutura Organizacional
5º	6.05.08	LA	Análise de Requisitos	5º	7.01.09	FN	Processos Organizacionais
6º	6.01.02	OC	Processo de Software	6º	7.01.03	FN	Diretrizes e Normas
6º	6.04.06	MP	Documentação	7º	7.01.06	FN	Alocação dos Profissionais
8º	6.05.10	LA	Codificação	8º	8.01.05	PT	Estrutura Organizacional
8º	6.05.13	LA	Manutenção	8º	8.01.01	PT	Missão da Organização
10º	4.01	DM	Domínio da Aplicação	10º	7.01.04	FN	Produtos e/ou Serviços
10º	6.04.20	MP	Solução de Problemas	11º	7.01.01	FN	Missão da Organização
10º	6.05.02	LA	Gerência de Projeto	12º	8.01.09	PT	Processos Organizacionais
13º	6.04.09	MP	Projeto (design)	12º	8.01.02	PT	Deficiências e Potenciais
14º	0.01.03	LT	MP da Indústria de Software	14º	7.01.07	FN	Distribuição de Competências
15º	6.05.09	LA	Projeto (design)	15º	8.01.07	PT	Distribuição de Competências
15º	0.01.02	LT	Inovações Tecnológicas	15º	7.01.02	FN	Deficiências e Potenciais
17º	6.05.16	LA	Melhoria de Processo	17º	6.04.14	MP	Engenharia Reversa e Reeng
18º	6.04.12	MP	Operação/Suporte a Usuários	17º	5.01.10	ORG	Relações Informais
18º	6.05.04	LA	Gerência de Qualidade	19º	6.04.19	MP	Processo de Fornecimento
18º	6.05.20	LA	Solução de Problemas	20º	9.01.07	CL	Distribuição de Competências
21º	0.01.04	LT	LA da Indústria de Software	21º	6.05.14	LA	Engenharia Reversa e Reeng
22º	6.06.01	DH	Gerência de Projeto	22º	9.01.01	CL	Missão da Organização
23º	6.05.15	LA	Reuso de Itens de Software	23º	6.05.05	LA	Gerência de Configuração
24º	6.05.17	LA	Treinamento	24º	6.04.03	MP	Gerência de Riscos
25º	9.01.02	CL	Deficiências e Potenciais	25º	6.05.03	LA	Gerência de Riscos

Tabela IV.15 – Tipos de Conhecimento quanto à Importância atribuída por Analistas com Média Experiência

Analista com Média Experiência (29 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.08	MP	Análise de Requisitos	1º	7.01.06	FN	Alocação dos Profissionais
1º	6.04.11	MP	Testes	2º	7.01.08	FN	Relações Informais
3º	6.04.10	MP	Codificação	2º	7.01.05	FN	Estrutura Organizacional
4º	6.04.09	MP	Projeto (design)	2º	7.01.01	FN	Missão da Organização
4º	6.04.20	MP	Solução de Problemas	5º	7.01.09	FN	Processos Organizacionais
6º	5.01.04	ORG	Objetivos e Metas	6º	7.01.03	FN	Diretrizes e Normas
6º	6.01.02	OC	Processo de Software	7º	8.01.03	PT	Diretrizes e Normas
8º	6.03	OC	Itens de Software	8º	8.01.08	PT	Relações Informais
8º	6.04.06	MP	Documentação	8º	7.01.07	FN	Distribuição de Competências
8º	6.05.10	LA	Codificação	10º	8.01.01	PT	Missão da Organização
11º	6.04.07	MP	Engenharia de Sistemas	11º	7.01.04	FN	Produtos e/ou Serviços
12º	6.04.02	MP	Gerência de Projeto	11º	7.01.02	FN	Deficiências e Potenciais
13º	6.05.11	LA	Testes	13º	8.01.09	PT	Processos Organizacionais
14º	6.02	OC	Roteiros de Documentos	14º	8.01.07	PT	Distribuição de Competências
15º	6.05.08	LA	Análise de Requisitos	14º	8.01.06	PT	Alocação dos Profissionais
16º	6.05.09	LA	Projeto (design)	14º	8.01.05	PT	Estrutura Organizacional
17º	0.01.01	LT	Conhecimento Teórico em ES	17º	8.01.02	PT	Deficiências e Potenciais
17º	0.01.02	LT	Inovações Tecnológicas	18º	6.05.18	LA	Processo de Aquisição
19º	5.01.07	ORG	Estrutura Organizacional	19º	9.01.08	CL	Relações Informais
19º	5.01.09	ORG	Distribuição de Competências	19º	6.04.18	MP	Processo de Aquisição
19º	6.06.01	DH	Gerência de Projeto	21º	6.04.19	MP	Processo de Fornecimento
22º	0.01.03	LT	MP da Indústria de Software	22º	6.05.19	LA	Processo de Fornecimento
22º	0.01.04	LT	LA da Indústria de Software	23º	5.01.10	ORG	Relações Informais
24º	9.01.09	CL	Processos Organizacionais	24º	6.04.14	MP	Engenharia Reversa e Reeng
25º	9.01.06	CL	Alocação dos Profissionais	25º	5.01.06	ORG	Produtos e/ou Serviços

Tabela IV.16 – Tipos de Conhecimento quanto à Importância atribuída por Analistas com Muita Experiência

Analista com Muita Experiência (11 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.04	MP	Gerência de Qualidade	1º	7.01.08	FN	Relações Informais
2º	4.01	DM	Domínio da Aplicação	2º	8.01.01	PT	Missão da Organização
3º	6.01.03	OC	Avaliação do Processo	3º	8.01.09	PT	Processos Organizacionais
3º	6.04.09	MP	Projeto (design)	3º	8.01.02	PT	Deficiências e Potenciais
3º	6.04.10	MP	Codificação	5º	8.01.03	PT	Diretrizes e Normas
6º	5.01.01	ORG	Missão da Organização	5º	7.01.09	FN	Processos Organizacionais
6º	5.01.02	ORG	Deficiências e Potenciais	5º	7.01.03	FN	Diretrizes e Normas
6º	5.01.05	ORG	Áreas Críticas para Objetivos	8º	8.01.08	PT	Relações Informais
6º	6.04.01	MP	Modelagem de Processo	8º	8.01.04	PT	Produtos e/ou Serviços
6º	6.04.02	MP	Gerência de Projeto	10º	7.01.05	FN	Estrutura Organizacional
6º	6.04.11	MP	Testes	10º	7.01.02	FN	Deficiências e Potenciais
6º	6.05.11	LA	Testes	10º	7.01.01	FN	Missão da Organização
13º	5.01.04	ORG	Objetivos e Metas	13º	8.01.06	PT	Alocação dos Profissionais
14º	5.01.11	ORG	Processos Organizacionais	13º	7.01.07	FN	Distribuição de Competências
14º	6.01.02	OC	Processo de Software	13º	7.01.04	FN	Produtos e/ou Serviços
14º	6.04.17	MP	Treinamento	16º	8.01.05	PT	Estrutura Organizacional
14º	6.05.10	LA	Codificação	16º	7.01.06	FN	Alocação dos Profissionais
18º	6.05.20	LA	Solução de Problemas	18º	6.05.19	LA	Processo de Fornecimento
19º	5.01.09	ORG	Distribuição de Competências	19º	5.01.10	ORG	Relações Informais
19º	6.04.03	MP	Gerência de Riscos	20º	5.01.06	ORG	Produtos e/ou Serviços
19º	6.04.06	MP	Documentação	21º	6.04.14	MP	Engenharia Reversa e Reeng
19º	6.04.20	MP	Solução de Problemas	21º	5.01.12	ORG	Contratos/Acordos
19º	6.05.04	LA	Gerência de Qualidade	23º	6.01.01	OC	Tipos de Software
19º	6.05.17	LA	Treinamento	24º	5.01.07	ORG	Estrutura Organizacional
19º	6.06.01	DH	Gerência de Projeto	25º	5.01.03	ORG	Diretrizes e Normas

Tabela IV.17 – Tipos de Conhecimento quanto à Importância atribuída por Analistas de Micro Empresas

Analista de Micro Empresa (2 participantes)							
Mais Importantes				Menos Importantes			
1º	6.01.01	OC	Tipos de Software	1º	7.01.09	FN	Processos Organizacionais
1º	6.01.02	OC	Processo de Software	1º	7.01.08	FN	Relações Informais
1º	6.04.06	MP	Documentação	1º	7.01.07	FN	Distribuição de Competências
4º	4.01	DM	Domínio da Aplicação	1º	7.01.06	FN	Alocação dos Profissionais
4º	6.01.03	OC	Avaliação do Processo	1º	7.01.05	FN	Estrutura Organizacional
4º	6.04.02	MP	Gerência de Projeto	1º	6.05.14	LA	Engenharia Reversa e Reeng
4º	6.04.07	MP	Engenharia de Sistemas	7º	0.01.04	LT	LA da Indústria de Software
4º	6.04.08	MP	Análise de Requisitos	7º	9.01.09	CL	Processos Organizacionais
4º	6.04.10	MP	Codificação	7º	9.01.03	CL	Diretrizes e Normas
4º	6.04.11	MP	Testes	7º	8.01.09	PT	Processos Organizacionais
4º	6.04.20	MP	Solução de Problemas	7º	8.01.08	PT	Relações Informais
4º	6.05.10	LA	Codificação	7º	8.01.07	PT	Distribuição de Competências
4º	6.05.11	LA	Testes	7º	8.01.06	PT	Alocação dos Profissionais
4º	6.05.12	LA	Operação/Suporte a Usuários	7º	8.01.03	PT	Diretrizes e Normas
4º	6.05.13	LA	Manutenção	7º	8.01.01	PT	Missão da Organização
16º	6.02	OC	Roteiros de Documentos	7º	7.01.03	FN	Diretrizes e Normas
16º	6.03	OC	Itens de Software	7º	7.01.01	FN	Missão da Organização
16º	6.04.04	MP	Gerência de Qualidade	7º	6.05.19	LA	Processo de Fornecimento
16º	6.04.05	MP	Gerência de Configuração	7º	6.05.18	LA	Processo de Aquisição
16º	6.04.09	MP	Projeto (design)	7º	6.04.18	MP	Processo de Aquisição
16º	6.04.12	MP	Operação/Suporte a Usuários	7º	6.04.14	MP	Engenharia Reversa e Reeng
16º	6.04.13	MP	Manutenção	7º	5.01.06	ORG	Produtos e/ou Serviços
16º	6.04.15	MP	Reuso de Itens de Software	23º	0.01.01	LT	Conhecimento Teórico em ES
16º	6.04.16	MP	Melhoria de Processo	23º	7.01.04	FN	Produtos e/ou Serviços
25º	6.04.19	MP	Processo de Fornecimento	23º	6.05.15	LA	Reuso de Itens de Software

Tabela IV.18 – Tipos de Conhecimento quanto à Importância atribuída por Analistas de Pequenas Empresas

Analista de Pequena Empresa (8 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.11	MP	Testes	1º	8.01.08	PT	Relações Informais
2º	5.01.09	ORG	Distribuição de Competências	1º	7.01.08	FN	Relações Informais
2º	6.04.04	MP	Gerência de Qualidade	3º	9.01.08	CL	Relações Informais
2º	6.04.06	MP	Documentação	3º	6.04.14	MP	Engenharia Reversa e Reeng
2º	6.04.07	MP	Engenharia de Sistemas	5º	7.01.09	FN	Processos Organizacionais
2º	6.04.08	MP	Análise de Requisitos	5º	7.01.05	FN	Estrutura Organizacional
2º	6.04.09	MP	Projeto (design)	7º	8.01.05	PT	Estrutura Organizacional
2º	6.04.10	MP	Codificação	8º	8.01.09	PT	Processos Organizacionais
2º	6.04.13	MP	Manutenção	8º	7.01.06	FN	Alocação dos Profissionais
2º	6.05.04	LA	Gerência de Qualidade	10º	8.01.03	PT	Diretrizes e Normas
2º	6.05.06	LA	Documentação	10º	8.01.02	PT	Deficiências e Potenciais
2º	6.05.10	LA	Codificação	10º	7.01.03	FN	Diretrizes e Normas
2º	6.05.11	LA	Testes	13º	8.01.07	PT	Distribuição de Competências
14º	5.01.05	ORG	Áreas Críticas para Objetivos	13º	8.01.06	PT	Alocação dos Profissionais
14º	6.01.02	OC	Processo de Software	13º	7.01.04	FN	Produtos e/ou Serviços
14º	6.01.03	OC	Avaliação do Processo	13º	7.01.01	FN	Missão da Organização
14º	6.04.01	MP	Modelagem de Processo	17º	8.01.04	PT	Produtos e/ou Serviços
14º	6.04.17	MP	Treinamento	17º	8.01.01	PT	Missão da Organização
14º	6.05.09	LA	Projeto (design)	17º	7.01.02	FN	Deficiências e Potenciais
20º	6.01.01	OC	Tipos de Software	20º	9.01.07	CL	Distribuição de Competências
20º	6.04.02	MP	Gerência de Projeto	20º	7.01.07	FN	Distribuição de Competências
20º	6.04.12	MP	Operação/Suporte a Usuários	22º	5.01.10	ORG	Relações Informais
20º	6.06.03	DH	Gerência de Qualidade	23º	5.01.06	ORG	Produtos e/ou Serviços
24º	5.01.07	ORG	Estrutura Organizacional	24º	4.01	DM	Domínio da Aplicação
25º	5.01.12	ORG	Contratos/Acordos	25º	0.01.03	LT	MP da Indústria de Software

Tabela IV.19 – Tipos de Conhecimento quanto à Importância atribuída por Analistas de Empresas Médias

Analista de Média Empresa (15 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.08	MP	Análise de Requisitos	1º	7.01.08	FN	Relações Informais
2º	6.04.09	MP	Projeto (design)	2º	7.01.07	FN	Distribuição de Competências
3º	6.01.02	OC	Processo de Software	3º	8.01.03	PT	Diretrizes e Normas
3º	6.04.10	MP	Codificação	3º	7.01.06	FN	Alocação dos Profissionais
3º	6.04.11	MP	Testes	3º	7.01.05	FN	Estrutura Organizacional
6º	6.01.03	OC	Avaliação do Processo	3º	7.01.01	FN	Missão da Organização
6º	6.04.04	MP	Gerência de Qualidade	7º	7.01.09	FN	Processos Organizacionais
6º	6.04.06	MP	Documentação	8º	8.01.08	PT	Relações Informais
9º	6.05.02	LA	Gerência de Projeto	9º	8.01.01	PT	Missão da Organização
10º	5.01.02	ORG	Deficiências e Potenciais	9º	7.01.03	FN	Diretrizes e Normas
10º	5.01.03	ORG	Diretrizes e Normas	11º	8.01.09	PT	Processos Organizacionais
10º	5.01.11	ORG	Processos Organizacionais	11º	8.01.05	PT	Estrutura Organizacional
10º	6.02	OC	Roteiros de Documentos	13º	8.01.02	PT	Deficiências e Potenciais
10º	6.05.10	LA	Codificação	13º	7.01.04	FN	Produtos e/ou Serviços
15º	6.04.02	MP	Gerência de Projeto	15º	8.01.07	PT	Distribuição de Competências
15º	6.05.08	LA	Análise de Requisitos	15º	8.01.06	PT	Alocação dos Profissionais
17º	5.01.04	ORG	Objetivos e Metas	15º	7.01.02	FN	Deficiências e Potenciais
17º	5.01.05	ORG	Áreas Críticas para Objetivos	18º	6.04.18	MP	Processo de Aquisição
17º	6.04.07	MP	Engenharia de Sistemas	19º	8.01.04	PT	Produtos e/ou Serviços
20º	6.04.20	MP	Solução de Problemas	20º	5.01.10	ORG	Relações Informais
21º	6.05.04	LA	Gerência de Qualidade	21º	6.04.14	MP	Engenharia Reversa e Reeng
22º	6.04.12	MP	Operação/Suporte a Usuários	22º	9.01.08	CL	Relações Informais
22º	6.04.17	MP	Treinamento	22º	9.01.07	CL	Distribuição de Competências
22º	6.05.01	LA	Modelagem de Processo	24º	9.01.06	CL	Alocação dos Profissionais
25º	6.05.03	LA	Gerência de Riscos	25º	5.01.06	ORG	Produtos e/ou Serviços

Tabela IV.20 – Tipos de Conhecimento quanto à Importância atribuída por Analistas de Grandes Empresas

Analista de Grande Empresa (45 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.11	MP	Testes	1º	7.01.08	FN	Relações Informais
2º	6.04.10	MP	Codificação	2º	8.01.03	PT	Diretrizes e Normas
3º	4.01	DM	Domínio da Aplicação	2º	7.01.03	FN	Diretrizes e Normas
4º	6.04.08	MP	Análise de Requisitos	4º	7.01.05	FN	Estrutura Organizacional
5º	6.05.11	LA	Testes	5º	7.01.09	FN	Processos Organizacionais
6º	5.01.04	ORG	Objetivos e Metas	5º	7.01.01	FN	Missão da Organização
6º	6.04.20	MP	Solução de Problemas	7º	8.01.08	PT	Relações Informais
8º	6.04.13	MP	Manutenção	7º	7.01.06	FN	Alocação dos Profissionais
9º	6.03	OC	Itens de Software	9º	8.01.01	PT	Missão da Organização
10º	6.04.09	MP	Projeto (design)	10º	7.01.04	FN	Produtos e/ou Serviços
10º	6.05.10	LA	Codificação	10º	7.01.02	FN	Deficiências e Potenciais
12º	6.01.02	OC	Processo de Software	12º	8.01.02	PT	Deficiências e Potenciais
12º	6.04.01	MP	Modelagem de Processo	13º	8.01.09	PT	Processos Organizacionais
14º	6.05.08	LA	Análise de Requisitos	14º	8.01.04	PT	Produtos e/ou Serviços
15º	6.04.06	MP	Documentação	15º	7.01.07	FN	Distribuição de Competências
16º	6.04.12	MP	Operação/Suporte a Usuários	16º	8.01.06	PT	Alocação dos Profissionais
17º	6.05.09	LA	Projeto (design)	16º	8.01.05	PT	Estrutura Organizacional
18º	6.04.04	MP	Gerência de Qualidade	18º	8.01.07	PT	Distribuição de Competências
19º	6.05.06	LA	Documentação	19º	9.01.08	CL	Relações Informais
20º	6.05.07	LA	Engenharia de Sistemas	20º	5.01.10	ORG	Relações Informais
21º	6.06.06	DH	Métricas de Processo	21º	6.05.18	LA	Processo de Aquisição
22º	5.01.09	ORG	Distribuição de Competências	22º	9.01.06	CL	Alocação dos Profissionais
23º	6.05.16	LA	Melhoria de Processo	23º	9.01.07	CL	Distribuição de Competências
24º	6.05.04	LA	Gerência de Qualidade	24º	6.04.14	MP	Engenharia Reversa e Reeng
24º	6.05.12	LA	Operação/Suporte a Usuários	25º	9.01.05	CL	Estrutura Organizacional

Tabela IV.21 – Tipos de Conhecimento quanto à Importância atribuída por Analistas considerando Projetos Pequenos

Analista considerando Projeto Pequeno (13 participantes)							
Mais Importantes				Menos Importantes			
1 ^o	6.04.11	MP	Testes	1 ^o	7.01.08	FN	Relações Informais
2 ^o	6.04.08	MP	Análise de Requisitos	2 ^o	8.01.08	PT	Relações Informais
2 ^o	6.04.09	MP	Projeto (design)	3 ^o	8.01.03	PT	Diretrizes e Normas
4 ^o	5.01.04	ORG	Objetivos e Metas	4 ^o	8.01.07	PT	Distribuição de Competências
4 ^o	6.04.10	MP	Codificação	4 ^o	7.01.07	FN	Distribuição de Competências
4 ^o	6.05.08	LA	Análise de Requisitos	6 ^o	8.01.09	PT	Processos Organizacionais
7 ^o	6.05.11	LA	Testes	6 ^o	8.01.05	PT	Estrutura Organizacional
8 ^o	6.03	OC	Itens de Software	6 ^o	8.01.04	PT	Produtos e/ou Serviços
8 ^o	6.04.04	MP	Gerência de Qualidade	6 ^o	8.01.01	PT	Missão da Organização
8 ^o	6.05.09	LA	Projeto (design)	6 ^o	7.01.09	FN	Processos Organizacionais
8 ^o	6.05.10	LA	Codificação	6 ^o	7.01.05	FN	Estrutura Organizacional
12 ^o	5.01.02	ORG	Deficiências e Potenciais	12 ^o	8.01.02	PT	Deficiências e Potenciais
12 ^o	6.04.06	MP	Documentação	12 ^o	7.01.03	FN	Diretrizes e Normas
12 ^o	6.04.13	MP	Manutenção	14 ^o	8.01.06	PT	Alocação dos Profissionais
15 ^o	4.01	DM	Domínio da Aplicação	15 ^o	7.01.06	FN	Alocação dos Profissionais
15 ^o	6.01.02	OC	Processo de Software	15 ^o	7.01.01	FN	Missão da Organização
15 ^o	6.02	OC	Roteiros de Documentos	17 ^o	7.01.04	FN	Produtos e/ou Serviços
18 ^o	6.04.02	MP	Gerência de Projeto	17 ^o	7.01.02	FN	Deficiências e Potenciais
18 ^o	6.04.12	MP	Operação/Suporte a Usuários	19 ^o	5.01.10	ORG	Relações Informais
18 ^o	6.04.20	MP	Solução de Problemas	20 ^o	9.01.08	CL	Relações Informais
18 ^o	6.05.04	LA	Gerência de Qualidade	21 ^o	9.01.07	CL	Distribuição de Competências
18 ^o	0.01.03	LT	MP da Indústria de Software	21 ^o	6.04.14	MP	Engenharia Reversa e Reeng
23 ^o	6.04.01	MP	Modelagem de Processo	23 ^o	9.01.06	CL	Alocação dos Profissionais
23 ^o	6.04.16	MP	Melhoria de Processo	23 ^o	9.01.05	CL	Estrutura Organizacional
25 ^o	6.05.01	LA	Modelagem de Processo	25 ^o	6.05.14	LA	Engenharia Reversa e Reeng

Tabela IV.22 – Tipos de Conhecimento quanto à Importância atribuída por Analistas considerando Projetos Médios

Analista considerando Projeto Médio (14 participantes)							
Mais Importantes				Menos Importantes			
1 ^o	6.04.08	MP	Análise de Requisitos	1 ^o	7.01.08	FN	Relações Informais
1 ^o	6.04.11	MP	Testes	2 ^o	7.01.05	FN	Estrutura Organizacional
3 ^o	6.04.10	MP	Codificação	3 ^o	8.01.01	PT	Missão da Organização
4 ^o	4.01	DM	Domínio da Aplicação	4 ^o	8.01.08	PT	Relações Informais
4 ^o	6.05.10	LA	Codificação	4 ^o	8.01.02	PT	Deficiências e Potenciais
6 ^o	6.04.06	MP	Documentação	4 ^o	7.01.06	FN	Alocação dos Profissionais
7 ^o	6.02	OC	Roteiros de Documentos	7 ^o	8.01.03	PT	Diretrizes e Normas
8 ^o	5.01.04	ORG	Objetivos e Metas	7 ^o	7.01.09	FN	Processos Organizacionais
8 ^o	6.01.03	OC	Avaliação do Processo	7 ^o	7.01.01	FN	Missão da Organização
8 ^o	6.04.02	MP	Gerência de Projeto	10 ^o	7.01.03	FN	Diretrizes e Normas
8 ^o	6.04.09	MP	Projeto (design)	11 ^o	7.01.04	FN	Produtos e/ou Serviços
8 ^o	6.04.13	MP	Manutenção	12 ^o	8.01.09	PT	Processos Organizacionais
8 ^o	6.05.02	LA	Gerência de Projeto	13 ^o	8.01.07	PT	Distribuição de Competências
14 ^o	6.04.01	MP	Modelagem de Processo	13 ^o	7.01.07	FN	Distribuição de Competências
14 ^o	6.04.04	MP	Gerência de Qualidade	13 ^o	7.01.02	FN	Deficiências e Potenciais
14 ^o	6.04.12	MP	Operação/Suporte a Usuários	16 ^o	8.01.05	PT	Estrutura Organizacional
14 ^o	0.01.03	LT	MP da Indústria de Software	16 ^o	6.05.19	LA	Processo de Fornecimento
18 ^o	0.01.04	LT	LA da Indústria de Software	18 ^o	8.01.06	PT	Alocação dos Profissionais
19 ^o	5.01.02	ORG	Deficiências e Potenciais	19 ^o	5.01.10	ORG	Relações Informais
19 ^o	6.05.09	LA	Projeto (design)	20 ^o	8.01.04	PT	Produtos e/ou Serviços
19 ^o	9.01.09	CL	Processos Organizacionais	21 ^o	6.04.14	MP	Engenharia Reversa e Reeng
22 ^o	5.01.03	ORG	Diretrizes e Normas	22 ^o	9.01.08	CL	Relações Informais
23 ^o	5.01.09	ORG	Distribuição de Competências	23 ^o	9.01.01	CL	Missão da Organização
23 ^o	6.05.01	LA	Modelagem de Processo	24 ^o	9.01.07	CL	Distribuição de Competências
23 ^o	6.05.03	LA	Gerência de Riscos	24 ^o	9.01.06	CL	Alocação dos Profissionais

Tabela IV.23 – Tipos de Conhecimento quanto à Importância atribuída por Analistas considerando Projetos Grandes

Analista considerando Projeto Grande (21 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.08	MP	Análise de Requisitos	1º	7.01.08	FN	Relações Informais
2º	6.04.06	MP	Documentação	2º	7.01.06	FN	Alocação dos Profissionais
3º	6.04.11	MP	Testes	2º	7.01.05	FN	Estrutura Organizacional
4º	6.01.02	OC	Processo de Software	2º	7.01.03	FN	Diretrizes e Normas
4º	6.04.02	MP	Gerência de Projeto	5º	7.01.09	FN	Processos Organizacionais
4º	6.04.09	MP	Projeto (design)	6º	7.01.01	FN	Missão da Organização
4º	6.04.10	MP	Codificação	7º	8.01.03	PT	Diretrizes e Normas
4º	6.05.08	LA	Análise de Requisitos	8º	8.01.08	PT	Relações Informais
9º	6.04.07	MP	Engenharia de Sistemas	9º	7.01.04	FN	Produtos e/ou Serviços
10º	6.05.09	LA	Projeto (design)	10º	7.01.07	FN	Distribuição de Competências
11º	5.01.09	ORG	Distribuição de Competências	11º	7.01.02	FN	Deficiências e Potenciais
12º	6.03	OC	Itens de Software	12º	8.01.01	PT	Missão da Organização
12º	6.04.20	MP	Solução de Problemas	13º	8.01.09	PT	Processos Organizacionais
12º	6.05.10	LA	Codificação	14º	8.01.05	PT	Estrutura Organizacional
15º	6.05.11	LA	Testes	15º	9.01.08	CL	Relações Informais
15º	0.01.02	LT	Inovações Tecnológicas	15º	6.05.18	LA	Processo de Aquisição
17º	5.01.04	ORG	Objetivos e Metas	17º	8.01.02	PT	Deficiências e Potenciais
17º	5.01.07	ORG	Estrutura Organizacional	17º	6.04.14	MP	Engenharia Reversa e Reeng
17º	6.06.01	DH	Gerência de Projeto	19º	8.01.06	PT	Alocação dos Profissionais
20º	0.01.04	LT	LA da Indústria de Software	20º	6.04.19	MP	Processo de Fornecimento
21º	0.01.01	LT	Conhecimento Teórico em ES	21º	8.01.07	PT	Distribuição de Competências
21º	0.01.03	LT	MP da Indústria de Software	21º	6.05.19	LA	Processo de Fornecimento
23º	6.02	OC	Roteiros de Documentos	21º	5.01.10	ORG	Relações Informais
24º	9.01.09	CL	Processos Organizacionais	24º	6.04.18	MP	Processo de Aquisição
25º	9.01.06	CL	Alocação dos Profissionais	25º	5.01.06	ORG	Produtos e/ou Serviços

Tabela IV.24 – Tipos de Conhecimento quanto à Importância atribuída por Analistas considerando Projetos de Manutenção

Analista considerando Projeto Manutenção (22 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.11	MP	Testes	1º	8.01.03	PT	Diretrizes e Normas
2º	6.04.10	MP	Codificação	2º	7.01.09	FN	Processos Organizacionais
3º	4.01	DM	Domínio da Aplicação	2º	7.01.01	FN	Missão da Organização
3º	6.01.02	OC	Processo de Software	4º	8.01.08	PT	Relações Informais
5º	6.04.20	MP	Solução de Problemas	4º	7.01.08	FN	Relações Informais
5º	6.05.11	LA	Testes	6º	7.01.06	FN	Alocação dos Profissionais
7º	5.01.04	ORG	Objetivos e Metas	6º	7.01.05	FN	Estrutura Organizacional
8º	5.01.01	ORG	Missão da Organização	6º	7.01.03	FN	Diretrizes e Normas
9º	6.05.10	LA	Codificação	9º	8.01.09	PT	Processos Organizacionais
10º	6.04.02	MP	Gerência de Projeto	10º	8.01.06	PT	Alocação dos Profissionais
11º	6.05.20	LA	Solução de Problemas	10º	8.01.01	PT	Missão da Organização
12º	6.01.01	OC	Tipos de Software	12º	8.01.05	PT	Estrutura Organizacional
12º	6.04.04	MP	Gerência de Qualidade	12º	8.01.02	PT	Deficiências e Potenciais
12º	6.04.09	MP	Projeto (design)	14º	7.01.02	FN	Deficiências e Potenciais
15º	5.01.05	ORG	Áreas Críticas para Objetivos	15º	7.01.07	FN	Distribuição de Competências
15º	6.04.01	MP	Modelagem de Processo	15º	7.01.04	FN	Produtos e/ou Serviços
15º	6.06.01	DH	Gerência de Projeto	17º	8.01.04	PT	Produtos e/ou Serviços
18º	5.01.09	ORG	Distribuição de Competências	18º	6.05.19	LA	Processo de Fornecimento
18º	5.01.11	ORG	Processos Organizacionais	19º	5.01.10	ORG	Relações Informais
20º	6.01.03	OC	Avaliação do Processo	20º	6.04.14	MP	Engenharia Reversa e Reeng
20º	6.04.06	MP	Documentação	21º	5.01.03	ORG	Diretrizes e Normas
22º	6.04.17	MP	Treinamento	21º	5.01.12	ORG	Contratos/Acordos
23º	6.05.04	LA	Gerência de Qualidade	21º	5.01.06	ORG	Produtos e/ou Serviços
24º	6.05.17	LA	Treinamento	21º	5.01.02	ORG	Deficiências e Potenciais
25º	6.04.03	MP	Gerência de Riscos	25º	5.01.07	ORG	Estrutura Organizacional

Tabela IV.25 – Tipos de Conhecimento quanto à Importância atribuída por Gerentes

Gerente (32 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.02	MP	Gerência de Projeto	1º	7.01.08	FN	Relações Informais
1º	6.04.11	MP	Testes	2º	7.01.09	FN	Processos Organizacionais
3º	6.04.04	MP	Gerência de Qualidade	2º	7.01.05	FN	Estrutura Organizacional
4º	6.04.08	MP	Análise de Requisitos	4º	7.01.06	FN	Alocação dos Profissionais
4º	6.04.09	MP	Projeto (design)	5º	7.01.01	FN	Missão da Organização
6º	6.04.01	MP	Modelagem de Processo	6º	8.01.08	PT	Relações Informais
6º	0.01.04	LT	LA da Indústria de Software	7º	7.01.03	FN	Diretrizes e Normas
8º	6.04.06	MP	Documentação	8º	8.01.05	PT	Estrutura Organizacional
9º	0.01.03	LT	MP da Indústria de Software	9º	8.01.06	PT	Alocação dos Profissionais
10º	6.04.07	MP	Engenharia de Sistemas	10º	8.01.09	PT	Processos Organizacionais
10º	6.04.20	MP	Solução de Problemas	10º	7.01.07	FN	Distribuição de Competências
12º	6.01.02	OC	Processo de Software	12º	8.01.07	PT	Distribuição de Competências
12º	6.04.10	MP	Codificação	12º	7.01.02	FN	Deficiências e Potenciais
12º	6.05.02	LA	Gerência de Projeto	14º	8.01.03	PT	Diretrizes e Normas
12º	6.06.03	DH	Gerência de Qualidade	15º	8.01.01	PT	Missão da Organização
16º	4.01	DM	Domínio da Aplicação	16º	8.01.02	PT	Deficiências e Potenciais
16º	6.05.08	LA	Análise de Requisitos	17º	8.01.04	PT	Produtos e/ou Serviços
16º	6.06.01	DH	Gerência de Projeto	18º	7.01.04	FN	Produtos e/ou Serviços
19º	5.01.09	ORG	Distribuição de Competências	19º	5.01.10	ORG	Relações Informais
20º	6.04.13	MP	Manutenção	20º	5.01.08	ORG	Alocação dos Profissionais
20º	6.05.20	LA	Solução de Problemas	21º	6.04.14	MP	Engenharia Reversa e Reeng
20º	6.06.05	DH	Métricas de Produto	22º	9.01.06	CL	Alocação dos Profissionais
23º	6.02	OC	Roteiros de Documentos	23º	5.01.12	ORG	Contratos/Acordos
23º	6.05.04	LA	Gerência de Qualidade	24º	9.01.04	CL	Produtos e/ou Serviços
25º	6.05.06	LA	Documentação	25º	6.01.03	OC	Avaliação do Processo

Tabela IV.26 – Tipos de Conhecimento quanto à Importância atribuída por Gerentes com Pouca Experiência

Gerente com Pouca Experiência (10 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.11	MP	Testes	1º	7.01.08	FN	Relações Informais
2º	6.04.06	MP	Documentação	2º	7.01.09	FN	Processos Organizacionais
3º	6.01.02	OC	Processo de Software	2º	7.01.06	FN	Alocação dos Profissionais
3º	6.04.02	MP	Gerência de Projeto	2º	7.01.01	FN	Missão da Organização
3º	6.04.04	MP	Gerência de Qualidade	5º	7.01.05	FN	Estrutura Organizacional
3º	6.04.10	MP	Codificação	6º	8.01.08	PT	Relações Informais
7º	5.01.09	ORG	Distribuição de Competências	7º	8.01.01	PT	Missão da Organização
7º	0.01.04	LT	LA da Indústria de Software	7º	7.01.07	FN	Distribuição de Competências
9º	6.03	OC	Ítem de Software	7º	7.01.04	FN	Produtos e/ou Serviços
9º	6.04.03	MP	Gerência de Riscos	7º	7.01.03	FN	Diretrizes e Normas
9º	0.01.02	LT	Inovações Tecnológicas	11º	8.01.09	PT	Processos Organizacionais
9º	0.01.03	LT	MP da Indústria de Software	11º	8.01.06	PT	Alocação dos Profissionais
13º	6.04.09	MP	Projeto (design)	11º	8.01.05	PT	Estrutura Organizacional
13º	6.04.16	MP	Melhoria de Processo	11º	8.01.03	PT	Diretrizes e Normas
13º	6.04.20	MP	Solução de Problemas	11º	8.01.02	PT	Deficiências e Potenciais
13º	6.05.02	LA	Gerência de Projeto	16º	8.01.04	PT	Produtos e/ou Serviços
17º	4.01	DM	Domínio da Aplicação	16º	7.01.02	FN	Deficiências e Potenciais
17º	6.05.20	LA	Solução de Problemas	18º	8.01.07	PT	Distribuição de Competências
17º	6.07	OC	Respostas para Perguntas	18º	6.05.18	LA	Processo de Aquisição
20º	6.05.01	LA	Modelagem de Processo	20º	6.04.19	MP	Processo de Fornecimento
20º	6.05.09	LA	Projeto (design)	20º	6.04.18	MP	Processo de Aquisição
22º	5.01.02	ORG	Deficiências e Potenciais	22º	5.01.10	ORG	Relações Informais
22º	6.01.01	OC	Tipos de Software	23º	5.01.07	ORG	Estrutura Organizacional
24º	5.01.01	ORG	Missão da Organização	24º	9.01.01	CL	Missão da Organização
24º	5.01.11	ORG	Processos Organizacionais	25º	9.01.06	CL	Alocação dos Profissionais

Tabela IV.27 – Tipos de Conhecimento quanto à Importância atribuída por Gerentes com Média Experiência

Gerente com Média Experiência (16 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.08	MP	Análise de Requisitos	1º	7.01.05	FN	Estrutura Organizacional
1º	6.04.09	MP	Projeto (design)	2º	7.01.09	FN	Processos Organizacionais
3º	6.04.02	MP	Gerência de Projeto	2º	7.01.08	FN	Relações Informais
3º	6.04.11	MP	Testes	4º	7.01.03	FN	Diretrizes e Normas
5º	6.04.04	MP	Gerência de Qualidade	4º	7.01.01	FN	Missão da Organização
5º	6.04.07	MP	Engenharia de Sistemas	6º	7.01.06	FN	Alocação dos Profissionais
5º	6.05.11	LA	Testes	7º	8.01.08	PT	Relações Informais
8º	6.04.20	MP	Solução de Problemas	8º	7.01.02	FN	Deficiências e Potenciais
8º	6.06.05	DH	Métricas de Produto	9º	8.01.06	PT	Alocação dos Profissionais
8º	0.01.04	LT	LA da Indústria de Software	9º	8.01.05	PT	Estrutura Organizacional
11º	6.05.02	LA	Gerência de Projeto	9º	7.01.07	FN	Distribuição de Competências
11º	6.06.03	DH	Gerência de Qualidade	12º	8.01.03	PT	Diretrizes e Normas
11º	0.01.03	LT	MP da Indústria de Software	13º	8.01.01	PT	Missão da Organização
14º	5.01.09	ORG	Distribuição de Competências	14º	8.01.02	PT	Deficiências e Potenciais
14º	6.04.01	MP	Modelagem de Processo	15º	7.01.04	FN	Produtos e/ou Serviços
14º	6.04.10	MP	Codificação	16º	8.01.04	PT	Produtos e/ou Serviços
14º	6.05.04	LA	Gerência de Qualidade	17º	9.01.08	CL	Relações Informais
14º	6.05.08	LA	Análise de Requisitos	18º	5.01.10	ORG	Relações Informais
14º	6.05.09	LA	Projeto (design)	19º	5.01.12	ORG	Contratos/Acordos
14º	6.06.02	DH	Gerência de Riscos	20º	9.01.04	CL	Produtos e/ou Serviços
21º	6.05.07	LA	Engenharia de Sistemas	21º	9.01.03	CL	Diretrizes e Normas
22º	6.03	OC	Itens de Software	22º	6.04.18	MP	Processo de Aquisição
22º	6.05.01	LA	Modelagem de Processo	22º	5.01.08	ORG	Alocação dos Profissionais
22º	6.05.20	LA	Solução de Problemas	24º	6.04.19	MP	Processo de Fornecimento
25º	6.05.06	LA	Documentação	25º	6.04.17	MP	Treinamento

Tabela IV.28 – Tipos de Conhecimento quanto à Importância atribuída por Gerentes com Muita Experiência

Gerente com Muita Experiência (6 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.01	MP	Modelagem de Processo	1º	7.01.08	FN	Relações Informais
2º	4.01	DM	Domínio da Aplicação	2º	8.01.08	PT	Relações Informais
2º	6.04.04	MP	Gerência de Qualidade	2º	7.01.06	FN	Alocação dos Profissionais
4º	6.01.02	OC	Processo de Software	4º	8.01.09	PT	Processos Organizacionais
4º	6.04.02	MP	Gerência de Projeto	4º	8.01.05	PT	Estrutura Organizacional
4º	6.04.08	MP	Análise de Requisitos	4º	7.01.05	FN	Estrutura Organizacional
4º	6.04.09	MP	Projeto (design)	7º	8.01.07	PT	Distribuição de Competências
4º	6.06.01	DH	Gerência de Projeto	7º	8.01.06	PT	Alocação dos Profissionais
4º	9.01.09	CL	Processos Organizacionais	7º	7.01.09	FN	Processos Organizacionais
10º	5.01.11	ORG	Processos Organizacionais	10º	8.01.04	PT	Produtos e/ou Serviços
10º	6.02	OC	Roteiros de Documentos	10º	7.01.03	FN	Diretrizes e Normas
10º	6.04.13	MP	Manutenção	12º	7.01.07	FN	Distribuição de Competências
10º	6.05.08	LA	Análise de Requisitos	13º	8.01.03	PT	Diretrizes e Normas
10º	6.06.03	DH	Gerência de Qualidade	13º	8.01.01	PT	Missão da Organização
10º	0.01.03	LT	MP da Indústria de Software	13º	7.01.01	FN	Missão da Organização
16º	5.01.01	ORG	Missão da Organização	16º	8.01.02	PT	Deficiências e Potenciais
16º	6.04.06	MP	Documentação	16º	7.01.02	FN	Deficiências e Potenciais
16º	6.05.04	LA	Gerência de Qualidade	18º	6.05.18	LA	Processo de Aquisição
16º	6.05.09	LA	Projeto (design)	18º	6.05.14	LA	Engenharia Reversa e Reeng
16º	6.05.13	LA	Manutenção	18º	5.01.10	ORG	Relações Informais
16º	6.05.15	LA	Reuso de Itens de Software	18º	5.01.08	ORG	Alocação dos Profissionais
16º	6.05.20	LA	Solução de Problemas	22º	7.01.04	FN	Produtos e/ou Serviços
23º	5.01.02	ORG	Deficiências e Potenciais	22º	6.04.18	MP	Processo de Aquisição
23º	6.05.06	LA	Documentação	24º	5.01.06	ORG	Produtos e/ou Serviços
23º	6.05.12	LA	Operação/Suporte a Usuários	25º	5.01.09	ORG	Distribuição de Competências

Tabela IV.29 – Tipos de Conhecimento quanto à Importância atribuída por Gerentes de Micro Empresas

Gerente de Micro Empresa (2 participantes)							
Mais Importantes				Menos Importantes			
1º	4.01	DM	Domínio da Aplicação	1º	5.01.12	ORG	Contratos/Acordos
1º	5.01.07	ORG	Estrutura Organizacional	1º	5.01.01	ORG	Missão da Organização
1º	6.01.01	OC	Tipos de Software	3º	5.01.10	ORG	Relações Informais
1º	6.01.02	OC	Processo de Software	3º	5.01.02	ORG	Deficiências e Potenciais
1º	6.04.04	MP	Gerência de Qualidade	5º	7.01.09	FN	Processos Organizacionais
1º	6.04.09	MP	Projeto (design)	5º	7.01.08	FN	Relações Informais
1º	6.04.10	MP	Codificação	5º	7.01.07	FN	Distribuição de Competências
1º	6.04.11	MP	Testes	5º	7.01.06	FN	Alocação dos Profissionais
1º	6.04.17	MP	Treinamento	5º	7.01.05	FN	Estrutura Organizacional
1º	6.04.20	MP	Solução de Problemas	5º	7.01.04	FN	Produtos e/ou Serviços
1º	6.05.19	LA	Processo de Fornecimento	5º	7.01.03	FN	Diretrizes e Normas
1º	6.05.20	LA	Solução de Problemas	5º	7.01.02	FN	Deficiências e Potenciais
13º	5.01.05	ORG	Áreas Críticas para Objetivos	5º	7.01.01	FN	Missão da Organização
13º	5.01.06	ORG	Produtos e/ou Serviços	5º	6.04.14	MP	Engenharia Reversa e Reeng
13º	5.01.09	ORG	Distribuição de Competências	5º	6.04.03	MP	Gerência de Riscos
13º	6.04.02	MP	Gerência de Projeto	5º	6.04.01	MP	Modelagem de Processo
13º	6.04.06	MP	Documentação	5º	6.01.03	OC	Avaliação do Processo
13º	6.05.04	LA	Gerência de Qualidade	5º	5.01.11	ORG	Processos Organizacionais
13º	6.05.10	LA	Codificação	5º	5.01.04	ORG	Objetivos e Metas
13º	6.05.11	LA	Testes	5º	5.01.03	ORG	Diretrizes e Normas
13º	6.05.17	LA	Treinamento	21º	8.01.09	PT	Processos Organizacionais
13º	6.06.01	DH	Gerência de Projeto	21º	8.01.08	PT	Relações Informais
13º	8.01.01	PT	Missão da Organização	21º	8.01.06	PT	Alocação dos Profissionais
13º	8.01.02	PT	Deficiências e Potenciais	21º	8.01.05	PT	Estrutura Organizacional
13º	8.01.03	PT	Diretrizes e Normas	21º	8.01.04	PT	Produtos e/ou Serviços

Tabela IV.30 – Tipos de Conhecimento quanto à Importância atribuída por Gerentes de Pequenas Empresas

Gerente de Pequena Empresa (2 participantes)							
Mais Importantes				Menos Importantes			
1º	6.03	OC	Itens de Software	1º	8.01.09	PT	Processos Organizacionais
1º	6.04.02	MP	Gerência de Projeto	1º	8.01.08	PT	Relações Informais
1º	6.04.04	MP	Gerência de Qualidade	1º	8.01.07	PT	Distribuição de Competências
1º	6.04.06	MP	Documentação	1º	8.01.03	PT	Diretrizes e Normas
1º	6.04.07	MP	Engenharia de Sistemas	1º	7.01.09	FN	Processos Organizacionais
1º	6.04.08	MP	Análise de Requisitos	1º	7.01.08	FN	Relações Informais
1º	6.04.09	MP	Projeto (design)	1º	7.01.07	FN	Distribuição de Competências
1º	6.04.10	MP	Codificação	1º	7.01.06	FN	Alocação dos Profissionais
1º	6.04.11	MP	Testes	9º	8.01.06	PT	Alocação dos Profissionais
1º	6.04.12	MP	Operação/Suporte a Usuários	9º	8.01.01	PT	Missão da Organização
1º	6.04.13	MP	Manutenção	9º	7.01.05	FN	Estrutura Organizacional
1º	6.04.15	MP	Reuso de Itens de Software	9º	7.01.04	FN	Produtos e/ou Serviços
1º	6.04.16	MP	Melhoria de Processo	9º	7.01.01	FN	Missão da Organização
1º	6.04.20	MP	Solução de Problemas	14º	9.01.09	CL	Processos Organizacionais
1º	0.01.04	LT	LA da Indústria de Software	14º	9.01.03	CL	Diretrizes e Normas
16º	6.01.03	OC	Avaliação do Processo	14º	7.01.03	FN	Diretrizes e Normas
16º	6.04.05	MP	Gerência de Configuração	14º	6.05.19	LA	Processo de Fornecimento
16º	0.01.01	LT	Conhecimento Teórico em ES	14º	6.05.18	LA	Processo de Aquisição
19º	4.01	DM	Domínio da Aplicação	14º	6.05.15	LA	Reuso de Itens de Software
19º	6.05.10	LA	Codificação	14º	6.05.14	LA	Engenharia Reversa e Reeng
19º	6.01.01	OC	Tipos de Software	14º	6.05.13	LA	Manutenção
19º	6.01.02	OC	Processo de Software	14º	6.04.18	MP	Processo de Aquisição
19º	6.02	OC	Roteiros de Documentos	14º	6.04.14	MP	Engenharia Reversa e Reeng
19º	6.05.11	LA	Testes	14º	5.01.06	ORG	Produtos e/ou Serviços
19º	6.05.12	LA	Operação/Suporte a Usuários	14º	6.04.19	MP	Processo de Fornecimento

Tabela IV.31 – Tipos de Conhecimento quanto à Importância atribuída por Gerentes de Empresas Médias

Gerente de Média Empresa (4 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.11	MP	Testes	1º	7.01.09	FN	Processos Organizacionais
2º	4.01	DM	Domínio da Aplicação	1º	7.01.08	FN	Relações Informais
2º	5.01.09	ORG	Distribuição de Competências	1º	7.01.06	FN	Alocação dos Profissionais
2º	6.04.02	MP	Gerência de Projeto	1º	7.01.05	FN	Estrutura Organizacional
2º	6.04.04	MP	Gerência de Qualidade	5º	7.01.01	FN	Missão da Organização
2º	6.04.08	MP	Análise de Requisitos	6º	8.01.08	PT	Relações Informais
2º	6.04.09	MP	Projeto (design)	6º	8.01.02	PT	Deficiências e Potenciais
2º	6.05.09	LA	Projeto (design)	6º	8.01.01	PT	Missão da Organização
2º	0.01.03	LT	MP da Indústria de Software	6º	7.01.07	FN	Distribuição de Competências
10º	6.01.02	OC	Processo de Software	6º	7.01.03	FN	Diretrizes e Normas
10º	6.01.03	OC	Avaliação do Processo	11º	8.01.09	PT	Processos Organizacionais
10º	6.04.01	MP	Modelagem de Processo	11º	8.01.05	PT	Estrutura Organizacional
10º	6.04.06	MP	Documentação	11º	8.01.04	PT	Produtos e/ou Serviços
10º	6.04.07	MP	Engenharia de Sistemas	11º	8.01.03	PT	Diretrizes e Normas
10º	6.04.10	MP	Codificação	11º	7.01.04	FN	Produtos e/ou Serviços
16º	5.01.12	ORG	Contratos/Acordos	11º	7.01.02	FN	Deficiências e Potenciais
16º	6.04.13	MP	Manutenção	17º	9.01.08	CL	Relações Informais
16º	6.04.17	MP	Treinamento	17º	8.01.06	PT	Alocação dos Profissionais
16º	6.05.11	LA	Testes	17º	6.04.14	MP	Engenharia Reversa e Reeng
16º	6.06.03	DH	Gerência de Qualidade	17º	5.01.10	ORG	Relações Informais
21º	5.01.05	ORG	Áreas Críticas para Objetivos	21º	9.01.07	CL	Distribuição de Competências
21º	6.04.12	MP	Operação/Suporte a Usuários	21º	8.01.07	PT	Distribuição de Competências
21º	6.05.04	LA	Gerência de Qualidade	21º	5.01.07	ORG	Estrutura Organizacional
21º	6.05.06	LA	Documentação	21º	5.01.06	ORG	Produtos e/ou Serviços
25º	6.01.01	OC	Tipos de Software	25º	6.05.10	LA	Codificação

Tabela IV.32 – Tipos de Conhecimento quanto à Importância atribuída por Gerentes de Grandes Empresas

Gerente de Grande Empresa (24 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.02	MP	Gerência de Projeto	1º	7.01.08	FN	Relações Informais
2º	6.04.04	MP	Gerência de Qualidade	2º	7.01.05	FN	Estrutura Organizacional
3º	6.04.08	MP	Análise de Requisitos	3º	7.01.09	FN	Processos Organizacionais
4º	6.04.06	MP	Documentação	4º	7.01.03	FN	Diretrizes e Normas
5º	6.03	OC	Itens de Software	5º	8.01.08	PT	Relações Informais
6º	6.01.02	OC	Processo de Software	5º	7.01.01	FN	Missão da Organização
6º	6.02	OC	Roteiros de Documentos	7º	8.01.05	PT	Estrutura Organizacional
6º	6.05.02	LA	Gerência de Projeto	8º	8.01.07	PT	Distribuição de Competências
9º	6.04.07	MP	Engenharia de Sistemas	9º	7.01.02	FN	Deficiências e Potenciais
9º	6.04.20	MP	Solução de Problemas	10º	5.01.01	ORG	Missão da Organização
11º	4.01	DM	Domínio da Aplicação	11º	7.01.04	FN	Produtos e/ou Serviços
11º	6.04.10	MP	Codificação	12º	5.01.10	ORG	Relações Informais
11º	6.05.04	LA	Gerência de Qualidade	13º	6.05.18	LA	Processo de Aquisição
11º	6.05.08	LA	Análise de Requisitos	14º	6.04.18	MP	Processo de Aquisição
15º	6.05.09	LA	Projeto (design)	15º	5.01.08	ORG	Alocação dos Profissionais
15º	6.06.05	DH	Métricas de Produto	16º	9.01.08	CL	Relações Informais
17º	6.04.16	MP	Melhoria de Processo	17º	5.01.12	ORG	Contratos/Acordos
17º	6.05.10	LA	Codificação	18º	9.01.06	CL	Alocação dos Profissionais
19º	6.06.06	DH	Métricas de Processo	18º	6.04.14	MP	Engenharia Reversa e Reeng
20º	6.05.06	LA	Documentação	20º	6.05.12	LA	Operação/Suporte a Usuários
21º	5.01.11	ORG	Processos Organizacionais	21º	5.01.07	ORG	Estrutura Organizacional
21º	6.04.15	MP	Reuso de Itens de Software	22º	6.01.03	OC	Avaliação do Processo
23º	5.01.05	ORG	Áreas Críticas para Objetivos	23º	6.04.17	MP	Treinamento
23º	6.01.01	OC	Tipos de Software	24º	5.01.04	ORG	Objetivos e Metas
25º	6.05.13	LA	Manutenção	24º	6.04.12	MP	Operação/Suporte a Usuários

Tabela IV.33 – Tipos de Conhecimento quanto à Importância atribuída por Gerentes considerando Projetos Pequenos

Gerente considerando Projeto Pequeno (4 participantes)							
Mais Importantes				Menos Importantes			
1 ^o	6.04.01	MP	Modelagem de Processo	1 ^o	8.01.05	PT	Estrutura Organizacional
1 ^o	6.04.04	MP	Gerência de Qualidade	1 ^o	7.01.06	FN	Alocação dos Profissionais
1 ^o	6.04.08	MP	Análise de Requisitos	3 ^o	8.01.06	PT	Alocação dos Profissionais
1 ^o	6.04.13	MP	Manutenção	3 ^o	7.01.08	FN	Relações Informais
5 ^o	6.01.03	OC	Avaliação do Processo	3 ^o	7.01.05	FN	Estrutura Organizacional
5 ^o	6.04.02	MP	Gerência de Projeto	6 ^o	8.01.09	PT	Processos Organizacionais
5 ^o	6.04.10	MP	Codificação	6 ^o	8.01.04	PT	Produtos e/ou Serviços
5 ^o	6.04.11	MP	Testes	6 ^o	7.01.07	FN	Distribuição de Competências
5 ^o	6.04.16	MP	Melhoria de Processo	6 ^o	7.01.03	FN	Diretrizes e Normas
5 ^o	6.04.20	MP	Solução de Problemas	6 ^o	7.01.01	FN	Missão da Organização
5 ^o	6.05.08	LA	Análise de Requisitos	11 ^o	7.01.09	FN	Processos Organizacionais
5 ^o	0.01.03	LT	MP da Indústria de Software	12 ^o	8.01.02	PT	Deficiências e Potenciais
13 ^o	5.01.05	ORG	Áreas Críticas para Objetivos	13 ^o	9.01.08	CL	Relações Informais
13 ^o	6.04.03	MP	Gerência de Riscos	13 ^o	6.05.19	LA	Processo de Fornecimento
13 ^o	6.04.17	MP	Treinamento	13 ^o	6.05.18	LA	Processo de Aquisição
13 ^o	6.05.10	LA	Codificação	13 ^o	5.01.12	ORG	Contratos/Acordos
13 ^o	6.05.20	LA	Solução de Problemas	13 ^o	5.01.10	ORG	Relações Informais
18 ^o	5.01.09	ORG	Distribuição de Competências	18 ^o	7.01.02	FN	Deficiências e Potenciais
18 ^o	6.04.12	MP	Operação/Suporte a Usuários	18 ^o	6.06.05	DH	Métricas de Produto
18 ^o	6.04.14	MP	Engenharia Reversa e Reeng	20 ^o	9.01.06	CL	Alocação dos Profissionais
18 ^o	6.06.01	DH	Gerência de Projeto	21 ^o	9.01.07	CL	Distribuição de Competências
18 ^o	9.01.03	CL	Diretrizes e Normas	22 ^o	6.05.14	LA	Engenharia Reversa e Reeng
18 ^o	0.01.01	LT	Conhecimento Teórico em ES	22 ^o	6.05.03	LA	Gerência de Riscos
18 ^o	0.01.02	LT	Inovações Tecnológicas	24 ^o	9.01.05	CL	Estrutura Organizacional
25 ^o	6.06.06	DH	Métricas de Processo	24 ^o	9.01.04	CL	Produtos e/ou Serviços

Tabela IV.34 – Tipos de Conhecimento quanto à Importância atribuída por Gerentes considerando Projetos Médios

Gerente considerando Projeto Médio (14 participantes)							
Mais Importantes				Menos Importantes			
1 ^o	6.04.11	MP	Testes	1 ^o	7.01.09	FN	Processos Organizacionais
2 ^o	6.04.02	MP	Gerência de Projeto	2 ^o	7.01.08	FN	Relações Informais
3 ^o	6.04.04	MP	Gerência de Qualidade	3 ^o	7.01.05	FN	Estrutura Organizacional
3 ^o	6.05.03	LA	Gerência de Riscos	4 ^o	7.01.01	FN	Missão da Organização
5 ^o	6.04.01	MP	Modelagem de Processo	5 ^o	7.01.03	FN	Diretrizes e Normas
5 ^o	6.04.06	MP	Documentação	6 ^o	8.01.09	PT	Processos Organizacionais
7 ^o	4.01	DM	Domínio da Aplicação	6 ^o	8.01.08	PT	Relações Informais
7 ^o	6.04.10	MP	Codificação	6 ^o	7.01.02	FN	Deficiências e Potenciais
9 ^o	0.01.02	LT	Inovações Tecnológicas	9 ^o	7.01.07	FN	Distribuição de Competências
10 ^o	6.02	OC	Roteiros de Documentos	10 ^o	8.01.05	PT	Estrutura Organizacional
10 ^o	6.04.07	MP	Engenharia de Sistemas	11 ^o	8.01.07	PT	Distribuição de Competências
10 ^o	6.04.20	MP	Solução de Problemas	11 ^o	8.01.06	PT	Alocação dos Profissionais
10 ^o	0.01.03	LT	MP da Indústria de Software	11 ^o	8.01.03	PT	Diretrizes e Normas
14 ^o	6.05.20	LA	Solução de Problemas	14 ^o	8.01.04	PT	Produtos e/ou Serviços
14 ^o	6.06.01	DH	Gerência de Projeto	14 ^o	8.01.02	PT	Deficiências e Potenciais
16 ^o	6.05.01	LA	Modelagem de Processo	14 ^o	5.01.10	ORG	Relações Informais
16 ^o	6.05.10	LA	Codificação	17 ^o	7.01.04	FN	Produtos e/ou Serviços
16 ^o	0.01.01	LT	Conhecimento Teórico em ES	18 ^o	9.01.01	CL	Missão da Organização
19 ^o	6.04.16	MP	Melhoria de Processo	18 ^o	6.05.14	LA	Engenharia Reversa e Reeng
19 ^o	6.05.16	LA	Melhoria de Processo	20 ^o	9.01.05	CL	Estrutura Organizacional
21 ^o	5.01.05	ORG	Áreas Críticas para Objetivos	20 ^o	6.05.15	LA	Reuso de Itens de Software
21 ^o	6.05.13	LA	Manutenção	22 ^o	6.04.14	MP	Engenharia Reversa e Reeng
21 ^o	6.06.02	DH	Gerência de Riscos	23 ^o	5.01.08	ORG	Alocação dos Profissionais
24 ^o	5.01.07	ORG	Estrutura Organizacional	24 ^o	6.04.19	MP	Processo de Fornecimento
24 ^o	6.04.03	MP	Gerência de Riscos	24 ^o	6.01.03	OC	Avaliação do Processo

Tabela IV.35 – Tipos de Conhecimento quanto à Importância atribuída por Gerentes considerando Projetos Grandes

Gerente considerando Projeto Grande (11 participantes)							
Mais Importantes				Menos Importantes			
1º	5.01.09	ORG	Distribuição de Competências	1º	8.01.09	PT	Processos Organizacionais
1º	6.04.04	MP	Gerência de Qualidade	2º	6.04.14	MP	Engenharia Reversa e Reeng
1º	6.04.08	MP	Análise de Requisitos	3º	6.04.19	MP	Processo de Fornecimento
4º	6.04.07	MP	Engenharia de Sistemas	3º	6.04.18	MP	Processo de Aquisição
4º	6.04.11	MP	Testes	5º	6.05.13	LA	Manutenção
4º	6.04.20	MP	Solução de Problemas	5º	5.01.10	ORG	Relações Informais
4º	6.05.02	LA	Gerência de Projeto	7º	5.01.08	ORG	Alocação dos Profissionais
8º	6.04.06	MP	Documentação	8º	9.01.06	CL	Alocação dos Profissionais
9º	6.04.10	MP	Codificação	9º	6.04.13	MP	Manutenção
9º	6.05.08	LA	Análise de Requisitos	10º	5.01.07	ORG	Estrutura Organizacional
9º	6.06.01	DH	Gerência de Projeto	11º	6.04.17	MP	Treinamento
9º	6.06.05	DH	Métricas de Produto	12º	9.01.01	CL	Missão da Organização
9º	6.06.06	DH	Métricas de Processo	12º	6.06.04	DH	Treinamento
14º	5.01.04	ORG	Objetivos e Metas	14º	5.01.06	ORG	Produtos e/ou Serviços
14º	6.03	OC	Itens de Software	15º	9.01.05	CL	Estrutura Organizacional
14º	6.06.03	DH	Gerência de Qualidade	15º	6.05.10	LA	Codificação
14º	6.01.01	OC	Tipos de Software	15º	6.05.06	LA	Documentação
18º	6.04.15	MP	Reuso de Itens de Software	15º	6.05.04	LA	Gerência de Qualidade
18º	6.04.16	MP	Melhoria de Processo	15º	5.01.11	ORG	Processos Organizacionais
18º	6.05.01	LA	Modelagem de Processo	15º	5.01.05	ORG	Áreas Críticas para Objetivos
18º	6.05.20	LA	Solução de Problemas	21º	9.01.02	CL	Deficiências e Potenciais
18º	6.07	OC	Respostas para Perguntas	21º	5.01.12	ORG	Contratos/Acordos
23º	4.01	DM	Domínio da Aplicação	21º	5.01.03	ORG	Diretrizes e Normas
23º	6.02	OC	Roteiros de Documentos	21º	5.01.02	ORG	Deficiências e Potenciais
23º	6.05.03	LA	Gerência de Riscos	25º	5.01.01	ORG	Missão da Organização

Tabela IV.36 – Tipos de Conhecimento quanto à Importância atribuída por Gerentes considerando Projetos de Manutenção

Gerente considerando Projeto Manutenção (3 participantes)							
Mais Importantes				Menos Importantes			
1º	4.01	DM	Domínio da Aplicação	1º	7.01.09	FN	Processos Organizacionais
2º	6.03	OC	Itens de Software	1º	7.01.08	FN	Relações Informais
2º	6.04.01	MP	Modelagem de Processo	1º	7.01.06	FN	Alocação dos Profissionais
2º	6.04.02	MP	Gerência de Projeto	4º	7.01.05	FN	Estrutura Organizacional
2º	6.04.09	MP	Projeto (design)	4º	7.01.03	FN	Diretrizes e Normas
6º	6.01.01	OC	Tipos de Software	4º	7.01.02	FN	Deficiências e Potenciais
6º	6.01.02	OC	Processo de Software	4º	7.01.01	FN	Missão da Organização
6º	6.04.04	MP	Gerência de Qualidade	4º	5.01.08	ORG	Alocação dos Profissionais
6º	6.04.08	MP	Análise de Requisitos	9º	8.01.01	PT	Missão da Organização
6º	6.04.16	MP	Melhoria de Processo	9º	7.01.07	FN	Distribuição de Competências
6º	6.05.02	LA	Gerência de Projeto	9º	7.01.04	FN	Produtos e/ou Serviços
6º	9.01.05	CL	Estrutura Organizacional	9º	5.01.10	ORG	Relações Informais
13º	5.01.11	ORG	Processos Organizacionais	13º	6.04.18	MP	Processo de Aquisição
13º	6.01.03	OC	Avaliação do Processo	13º	5.01.12	ORG	Contratos/Acordos
13º	6.04.07	MP	Engenharia de Sistemas	13º	5.01.02	ORG	Deficiências e Potenciais
13º	6.04.11	MP	Testes	16º	5.01.07	ORG	Estrutura Organizacional
13º	6.04.15	MP	Reuso de Itens de Software	16º	5.01.03	ORG	Diretrizes e Normas
13º	6.04.19	MP	Processo de Fornecimento	18º	9.01.08	CL	Relações Informais
13º	6.05.04	LA	Gerência de Qualidade	18º	9.01.07	CL	Distribuição de Competências
13º	9.01.06	CL	Alocação dos Profissionais	18º	9.01.04	CL	Produtos e/ou Serviços
21º	6.02	OC	Roteiros de Documentos	18º	9.01.03	CL	Diretrizes e Normas
21º	6.04.06	MP	Documentação	18º	9.01.02	CL	Deficiências e Potenciais
21º	6.04.10	MP	Codificação	18º	9.01.01	CL	Missão da Organização
21º	6.05.05	LA	Gerência de Configuração	18º	6.05.06	LA	Documentação
21º	6.05.01	LA	Modelagem de Processo	18º	6.04.14	MP	Engenharia Reversa e Reeng

Tabela IV.37 – Tipos de Conhecimento quanto à Importância atribuída por Diretores

Diretor (10 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.10	MP	Codificação	1º	7.01.08	FN	Relações Informais
1º	6.04.11	MP	Testes	2º	7.01.06	FN	Alocação dos Profissionais
3º	6.04.16	MP	Melhoria de Processo	2º	6.04.14	MP	Engenharia Reversa e Reeng
3º	6.06.01	DH	Gerência de Projeto	2º	5.01.07	ORG	Estrutura Organizacional
5º	4.01	DM	Domínio da Aplicação	5º	7.01.09	FN	Processos Organizacionais
5º	6.01.02	OC	Processo de Software	5º	7.01.05	FN	Estrutura Organizacional
5º	6.01.03	OC	Avaliação do Processo	5º	7.01.03	FN	Diretrizes e Normas
5º	6.04.02	MP	Gerência de Projeto	8º	7.01.02	FN	Deficiências e Potenciais
5º	6.04.04	MP	Gerência de Qualidade	8º	7.01.01	FN	Missão da Organização
5º	6.04.06	MP	Documentação	10º	6.05.14	LA	Engenharia Reversa e Reeng
5º	6.04.20	MP	Solução de Problemas	11º	7.01.07	FN	Distribuição de Competências
5º	6.05.08	LA	Análise de Requisitos	11º	5.01.12	ORG	Contratos/Acordos
5º	6.05.20	LA	Solução de Problemas	13º	8.01.08	PT	Relações Informais
5º	6.06.03	DH	Gerência de Qualidade	13º	6.05.03	LA	Gerência de Riscos
5º	6.06.05	DH	Métricas de Produto	15º	8.01.05	PT	Estrutura Organizacional
5º	6.06.06	DH	Métricas de Processo	16º	8.01.09	PT	Processos Organizacionais
17º	6.04.01	MP	Modelagem de Processo	16º	5.01.11	ORG	Processos Organizacionais
17º	6.04.07	MP	Engenharia de Sistemas	16º	5.01.10	ORG	Relações Informais
17º	6.04.17	MP	Treinamento	19º	9.01.06	CL	Alocação dos Profissionais
20º	6.01.01	OC	Tipos de Software	20º	6.06.02	DH	Gerência de Riscos
20º	6.03	OC	Itens de Software	20º	6.05.15	LA	Reuso de Itens de Software
20º	6.05.02	LA	Gerência de Projeto	20º	6.05.01	LA	Modelagem de Processo
23º	6.04.08	MP	Análise de Requisitos	23º	9.01.05	CL	Estrutura Organizacional
23º	6.05.10	LA	Codificação	24º	6.05.13	LA	Manutenção
23º	6.07	OC	Respostas para Perguntas	25º	6.02	OC	Roteiros de Documentos

Tabela IV.38 – Tipos de Conhecimento quanto à Importância atribuída por Diretores com Pouca Experiência

Diretor com Pouca Experiência (2 participantes)							
Mais Importantes				Menos Importantes			
1º	4.01	DM	Domínio da Aplicação	1º	6.05.15	LA	Reuso de Itens de Software
1º	6.04.11	MP	Testes	1º	6.05.14	LA	Engenharia Reversa e Reeng
1º	6.05.11	LA	Testes	1º	6.05.13	LA	Manutenção
1º	6.05.18	LA	Processo de Aquisição	1º	6.04.15	MP	Reuso de Itens de Software
1º	6.05.20	LA	Solução de Problemas	1º	6.04.14	MP	Engenharia Reversa e Reeng
1º	9.01.01	CL	Missão da Organização	6º	0.01.02	LT	Inovações Tecnológicas
1º	9.01.05	CL	Estrutura Organizacional	6º	0.01.01	LT	Conhecimento Teórico em ES
1º	9.01.06	CL	Alocação dos Profissionais	6º	7.01.08	FN	Relações Informais
1º	9.01.08	CL	Relações Informais	6º	7.01.07	FN	Distribuição de Competências
1º	9.01.09	CL	Processos Organizacionais	6º	7.01.06	FN	Alocação dos Profissionais
11º	5.01.01	ORG	Missão da Organização	6º	7.01.03	FN	Diretrizes e Normas
11º	5.01.04	ORG	Objetivos e Metas	6º	7.01.02	FN	Deficiências e Potenciais
11º	5.01.09	ORG	Distribuição de Competências	6º	6.06.05	DH	Métricas de Produto
11º	5.01.12	ORG	Contratos/Acordos	6º	6.05.08	LA	Análise de Requisitos
11º	6.01.01	OC	Tipos de Software	6º	6.05.07	LA	Engenharia de Sistemas
11º	6.01.02	OC	Processo de Software	6º	6.05.01	LA	Modelagem de Processo
11º	6.04.10	MP	Codificação	6º	6.04.13	MP	Manutenção
11º	6.04.12	MP	Operação/Suporte a Usuários	6º	6.04.08	MP	Análise de Requisitos
11º	6.04.16	MP	Melhoria de Processo	6º	6.04.03	MP	Gerência de Riscos
11º	6.04.17	MP	Treinamento	6º	6.03	OC	Itens de Software
11º	6.04.18	MP	Processo de Aquisição	6º	5.01.07	ORG	Estrutura Organizacional
11º	6.04.19	MP	Processo de Fornecimento	22º	7.01.05	FN	Estrutura Organizacional
11º	6.05.02	LA	Gerência de Projeto	22º	6.05.10	LA	Codificação
11º	6.05.12	LA	Operação/Suporte a Usuários	22º	5.01.11	ORG	Processos Organizacionais
11º	6.05.19	LA	Processo de Fornecimento	25º	6.01.03	OC	Avaliação do Processo

Tabela IV.39 – Tipos de Conhecimento quanto à Importância atribuída por Diretores com Média Experiência

Diretor com Média Experiência (5 participantes)							
Mais Importantes				Menos Importantes			
1º	6.03	OC	Itens de Software	1º	7.01.08	FN	Relações Informais
1º	6.04.10	MP	Codificação	2º	7.01.09	FN	Processos Organizacionais
1º	6.05.10	LA	Codificação	3º	7.01.03	FN	Diretrizes e Normas
4º	4.01	DM	Domínio da Aplicação	3º	5.01.10	ORG	Relações Informais
4º	6.04.04	MP	Gerência de Qualidade	5º	7.01.05	FN	Estrutura Organizacional
4º	6.04.15	MP	Reuso de Itens de Software	5º	7.01.02	FN	Deficiências e Potenciais
4º	6.05.08	LA	Análise de Requisitos	5º	7.01.01	FN	Missão da Organização
4º	6.06.05	DH	Métricas de Produto	5º	5.01.12	ORG	Contratos/Acordos
4º	6.06.06	DH	Métricas de Processo	5º	5.01.07	ORG	Estrutura Organizacional
10º	6.01.01	OC	Tipos de Software	10º	9.01.08	CL	Relações Informais
10º	6.01.02	OC	Processo de Software	10º	8.01.08	PT	Relações Informais
10º	6.01.03	OC	Avaliação do Processo	10º	5.01.08	ORG	Alocação dos Profissionais
10º	6.04.02	MP	Gerência de Projeto	13º	7.01.04	FN	Produtos e/ou Serviços
10º	6.04.06	MP	Documentação	13º	6.04.18	MP	Processo de Aquisição
10º	6.04.07	MP	Engenharia de Sistemas	15º	8.01.05	PT	Estrutura Organizacional
10º	6.04.08	MP	Análise de Requisitos	15º	6.04.14	MP	Engenharia Reversa e Reeng
10º	6.04.16	MP	Melhoria de Processo	17º	9.01.06	CL	Alocação dos Profissionais
10º	6.04.20	MP	Solução de Problemas	17º	8.01.07	PT	Distribuição de Competências
10º	6.05.02	LA	Gerência de Projeto	17º	6.05.18	LA	Processo de Aquisição
10º	6.05.06	LA	Documentação	17º	5.01.11	ORG	Processos Organizacionais
10º	6.05.13	LA	Manutenção	17º	5.01.05	ORG	Áreas Críticas para Objetivos
22º	6.04.17	MP	Treinamento	17º	5.01.04	ORG	Objetivos e Metas
23º	6.02	OC	Roteiros de Documentos	17º	5.01.01	ORG	Missão da Organização
23º	6.05.04	LA	Gerência de Qualidade	24º	6.05.12	LA	Operação/Suporte a Usuários
23º	6.05.09	LA	Projeto (design)	24º	6.04.12	MP	Operação/Suporte a Usuários

Tabela IV.40 – Tipos de Conhecimento quanto à Importância atribuída por Diretores com Muita Experiência

Diretor com Muita Experiência (3 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.11	MP	Testes	1º	6.05.14	LA	Engenharia Reversa e Reeng
2º	5.01.10	ORG	Relações Informais	2º	6.04.14	MP	Engenharia Reversa e Reeng
2º	6.04.01	MP	Modelagem de Processo	3º	7.01.08	FN	Relações Informais
2º	6.04.12	MP	Operação/Suporte a Usuários	3º	7.01.06	FN	Alocação dos Profissionais
2º	6.04.13	MP	Manutenção	3º	7.01.05	FN	Estrutura Organizacional
2º	6.05.08	LA	Análise de Requisitos	3º	7.01.01	FN	Missão da Organização
2º	6.06.01	DH	Gerência de Projeto	3º	6.05.03	LA	Gerência de Riscos
2º	6.06.05	DH	Métricas de Produto	8º	6.05.19	LA	Processo de Fornecimento
2º	8.01.02	PT	Deficiências e Potenciais	8º	6.05.18	LA	Processo de Aquisição
2º	8.01.04	PT	Produtos e/ou Serviços	8º	5.01.12	ORG	Contratos/Acordos
11º	5.01.05	ORG	Áreas Críticas para Objetivos	11º	9.01.08	CL	Relações Informais
11º	5.01.09	ORG	Distribuição de Competências	11º	9.01.06	CL	Alocação dos Profissionais
11º	6.01.03	OC	Avaliação do Processo	11º	8.01.05	PT	Estrutura Organizacional
11º	6.04.02	MP	Gerência de Projeto	11º	7.01.09	FN	Processos Organizacionais
11º	6.04.03	MP	Gerência de Riscos	11º	7.01.03	FN	Diretrizes e Normas
11º	6.04.04	MP	Gerência de Qualidade	11º	7.01.02	FN	Deficiências e Potenciais
11º	6.04.10	MP	Codificação	11º	6.05.10	LA	Codificação
11º	6.04.16	MP	Melhoria de Processo	18º	9.01.05	CL	Estrutura Organizacional
11º	6.04.20	MP	Solução de Problemas	18º	8.01.09	PT	Processos Organizacionais
11º	9.01.03	CL	Diretrizes e Normas	18º	8.01.06	PT	Alocação dos Profissionais
11º	0.01.01	LT	Conhecimento Teórico em ES	18º	7.01.07	FN	Distribuição de Competências
11º	0.01.02	LT	Inovações Tecnológicas	18º	6.05.20	LA	Solução de Problemas
11º	0.01.03	LT	MP da Indústria de Software	23º	9.01.07	CL	Distribuição de Competências
24º	6.04.08	MP	Análise de Requisitos	23º	9.01.04	CL	Produtos e/ou Serviços
24º	6.04.17	MP	Treinamento	23º	6.06.06	DH	Métricas de Processo

Tabela IV.41 – Tipos de Conhecimento quanto à Importância atribuída por Diretores de Micro Empresas

Diretor de Micro Empresa (5 participantes)							
Mais Importantes				Menos Importantes			
1 ^o	6.04.11	MP	Testes	1 ^o	7.01.08	FN	Relações Informais
2 ^o	6.04.10	MP	Codificação	1 ^o	7.01.05	FN	Estrutura Organizacional
2 ^o	8.01.02	PT	Deficiências e Potenciais	1 ^o	6.05.14	LA	Engenharia Reversa e Reeng
4 ^o	4.01	DM	Domínio da Aplicação	4 ^o	5.01.08	ORG	Alocação dos Profissionais
4 ^o	6.01.03	OC	Avaliação do Processo	4 ^o	5.01.07	ORG	Estrutura Organizacional
4 ^o	6.04.02	MP	Gerência de Projeto	6 ^o	7.01.07	FN	Distribuição de Competências
4 ^o	6.04.19	MP	Processo de Fornecimento	6 ^o	7.01.03	FN	Diretrizes e Normas
4 ^o	6.04.20	MP	Solução de Problemas	6 ^o	7.01.02	FN	Deficiências e Potenciais
4 ^o	6.06.01	DH	Gerência de Projeto	6 ^o	7.01.01	FN	Missão da Organização
4 ^o	9.01.01	CL	Missão da Organização	6 ^o	6.05.03	LA	Gerência de Riscos
11 ^o	5.01.10	ORG	Relações Informais	6 ^o	6.04.14	MP	Engenharia Reversa e Reeng
11 ^o	6.04.01	MP	Modelagem de Processo	12 ^o	7.01.09	FN	Processos Organizacionais
11 ^o	6.04.04	MP	Gerência de Qualidade	12 ^o	6.05.01	LA	Modelagem de Processo
11 ^o	6.04.06	MP	Documentação	14 ^o	8.01.05	PT	Estrutura Organizacional
11 ^o	6.04.07	MP	Engenharia de Sistemas	14 ^o	6.05.15	LA	Reuso de Itens de Software
11 ^o	6.04.16	MP	Melhoria de Processo	14 ^o	6.05.13	LA	Manutenção
11 ^o	6.05.20	LA	Solução de Problemas	17 ^o	8.01.09	PT	Processos Organizacionais
11 ^o	8.01.03	PT	Diretrizes e Normas	17 ^o	8.01.08	PT	Relações Informais
11 ^o	8.01.04	PT	Produtos e/ou Serviços	17 ^o	6.06.02	DH	Gerência de Riscos
11 ^o	0.01.03	LT	MP da Indústria de Software	17 ^o	6.05.16	LA	Melhoria de Processo
21 ^o	5.01.05	ORG	Áreas Críticas para Objetivos	21 ^o	7.01.04	FN	Produtos e/ou Serviços
21 ^o	8.01.07	PT	Distribuição de Competências	21 ^o	6.05.10	LA	Codificação
21 ^o	9.01.05	CL	Estrutura Organizacional	21 ^o	6.04.03	MP	Gerência de Riscos
24 ^o	0.01.01	LT	Conhecimento Teórico em ES	21 ^o	6.02	OC	Roteiros de Documentos
24 ^o	0.01.02	LT	Inovações Tecnológicas	25 ^o	8.01.06	PT	Alocação dos Profissionais

Tabela IV.42 – Tipos de Conhecimento quanto à Importância atribuída por Diretores de Pequenas Empresas

Diretor de Pequena Empresa (3 participantes)							
Mais Importantes				Menos Importantes			
1 ^o	6.04.10	MP	Codificação	1 ^o	6.04.14	MP	Engenharia Reversa e Reeng
1 ^o	6.04.16	MP	Melhoria de Processo	2 ^o	5.01.08	ORG	Alocação dos Profissionais
1 ^o	6.04.17	MP	Treinamento	2 ^o	5.01.07	ORG	Estrutura Organizacional
1 ^o	6.05.08	LA	Análise de Requisitos	4 ^o	8.01.09	PT	Processos Organizacionais
1 ^o	6.05.10	LA	Codificação	4 ^o	6.04.18	MP	Processo de Aquisição
1 ^o	6.05.13	LA	Manutenção	4 ^o	5.01.12	ORG	Contratos/Acordos
1 ^o	6.06.05	DH	Métricas de Produto	4 ^o	5.01.11	ORG	Processos Organizacionais
1 ^o	6.06.06	DH	Métricas de Processo	4 ^o	5.01.10	ORG	Relações Informais
9 ^o	6.02	OC	Roteiros de Documentos	4 ^o	5.01.06	ORG	Produtos e/ou Serviços
9 ^o	6.03	OC	Itens de Software	4 ^o	5.01.05	ORG	Áreas Críticas para Objetivos
9 ^o	6.04.04	MP	Gerência de Qualidade	4 ^o	5.01.04	ORG	Objetivos e Metas
9 ^o	6.04.06	MP	Documentação	4 ^o	5.01.03	ORG	Diretrizes e Normas
9 ^o	6.04.11	MP	Testes	13 ^o	9.01.06	CL	Alocação dos Profissionais
9 ^o	6.04.13	MP	Manutenção	13 ^o	9.01.01	CL	Missão da Organização
9 ^o	6.04.15	MP	Reuso de Itens de Software	13 ^o	6.05.03	LA	Gerência de Riscos
9 ^o	6.04.20	MP	Solução de Problemas	13 ^o	5.01.02	ORG	Deficiências e Potenciais
9 ^o	6.05.01	LA	Modelagem de Processo	13 ^o	5.01.01	ORG	Missão da Organização
9 ^o	6.05.02	LA	Gerência de Projeto	18 ^o	9.01.05	CL	Estrutura Organizacional
9 ^o	6.05.06	LA	Documentação	18 ^o	9.01.02	CL	Deficiências e Potenciais
9 ^o	6.05.20	LA	Solução de Problemas	18 ^o	6.05.04	LA	Gerência de Qualidade
9 ^o	6.06.01	DH	Gerência de Projeto	18 ^o	6.04.19	MP	Processo de Fornecimento
9 ^o	6.06.03	DH	Gerência de Qualidade	18 ^o	6.04.08	MP	Análise de Requisitos
9 ^o	6.06.04	DH	Treinamento	18 ^o	6.01.01	OC	Tipos de Software
9 ^o	6.07	OC	Respostas para Perguntas	18 ^o	5.01.09	ORG	Distribuição de Competências
25 ^o	4.01	DM	Domínio da Aplicação	25 ^o	6.04.07	MP	Engenharia de Sistemas

Tabela IV.43 – Tipos de Conhecimento quanto à Importância atribuída por Diretores de Empresas Médias

Diretor de Média Empresa (2 participantes)							
Mais Importantes				Menos Importantes			
1º	4.01	DM	Domínio da Aplicação	1º	5.01.12	ORG	Contratos/Acordos
1º	5.01.02	ORG	Deficiências e Potenciais	2º	7.01.08	FN	Relações Informais
1º	5.01.03	ORG	Diretrizes e Normas	2º	5.01.10	ORG	Relações Informais
1º	5.01.11	ORG	Processos Organizacionais	4º	7.01.09	FN	Processos Organizacionais
1º	6.01.01	OC	Tipos de Software	5º	9.01.08	CL	Relações Informais
1º	6.01.02	OC	Processo de Software	5º	6.04.14	MP	Engenharia Reversa e Reeng
1º	6.01.03	OC	Avaliação do Processo	5º	5.01.07	ORG	Estrutura Organizacional
1º	6.02	OC	Roteiros de Documentos	8º	9.01.07	CL	Distribuição de Competências
1º	6.03	OC	Itens de Software	8º	9.01.06	CL	Alocação dos Profissionais
1º	6.04.01	MP	Modelagem de Processo	8º	9.01.05	CL	Estrutura Organizacional
1º	6.04.02	MP	Gerência de Projeto	8º	8.01.01	PT	Missão da Organização
1º	6.04.04	MP	Gerência de Qualidade	8º	7.01.05	FN	Estrutura Organizacional
1º	6.04.06	MP	Documentação	8º	7.01.03	FN	Diretrizes e Normas
1º	6.04.07	MP	Engenharia de Sistemas	8º	7.01.02	FN	Deficiências e Potenciais
1º	6.04.08	MP	Análise de Requisitos	8º	7.01.01	FN	Missão da Organização
1º	6.04.09	MP	Projeto (design)	8º	6.04.19	MP	Processo de Fornecimento
1º	6.04.10	MP	Codificação	8º	5.01.08	ORG	Alocação dos Profissionais
1º	6.04.11	MP	Testes	18º	9.01.04	CL	Produtos e/ou Serviços
1º	6.04.15	MP	Reuso de Itens de Software	18º	9.01.03	CL	Diretrizes e Normas
1º	6.04.16	MP	Melhoria de Processo	18º	9.01.02	CL	Deficiências e Potenciais
1º	6.05.01	LA	Modelagem de Processo	18º	9.01.01	CL	Missão da Organização
1º	6.05.02	LA	Gerência de Projeto	18º	7.01.07	FN	Distribuição de Competências
1º	6.05.04	LA	Gerência de Qualidade	18º	7.01.06	FN	Alocação dos Profissionais
1º	6.05.05	LA	Gerência de Configuração	18º	7.01.04	FN	Produtos e/ou Serviços
1º	6.05.06	LA	Documentação	18º	6.04.18	MP	Processo de Aquisição

Tabela IV.44 – Tipos de Conhecimento quanto à Importância atribuída por Diretores considerando Projetos Médios

Diretor considerando Projeto Médio (7 participantes)							
Mais Importantes				Menos Importantes			
1º	6.04.11	MP	Testes	1º	7.01.08	FN	Relações Informais
2º	8.01.02	PT	Deficiências e Potenciais	1º	6.05.14	LA	Engenharia Reversa e Reeng
3º	4.01	DM	Domínio da Aplicação	1º	5.01.07	ORG	Estrutura Organizacional
3º	6.04.01	MP	Modelagem de Processo	4º	6.04.14	MP	Engenharia Reversa e Reeng
3º	6.04.12	MP	Operação/Suporte a Usuários	4º	5.01.08	ORG	Alocação dos Profissionais
3º	6.04.16	MP	Melhoria de Processo	6º	7.01.09	FN	Processos Organizacionais
3º	6.04.20	MP	Solução de Problemas	6º	7.01.06	FN	Alocação dos Profissionais
3º	6.05.20	LA	Solução de Problemas	8º	7.01.05	FN	Estrutura Organizacional
3º	9.01.01	CL	Missão da Organização	8º	6.05.03	LA	Gerência de Riscos
3º	9.01.03	CL	Diretrizes e Normas	10º	7.01.03	FN	Diretrizes e Normas
3º	0.01.03	LT	MP da Indústria de Software	10º	7.01.02	FN	Deficiências e Potenciais
3º	6.06.01	DH	Gerência de Projeto	10º	5.01.12	ORG	Contratos/Acordos
13º	6.01.03	OC	Avaliação do Processo	13º	8.01.05	PT	Estrutura Organizacional
13º	6.04.04	MP	Gerência de Qualidade	13º	6.05.01	LA	Modelagem de Processo
13º	6.04.17	MP	Treinamento	13º	6.04.18	MP	Processo de Aquisição
13º	6.05.08	LA	Análise de Requisitos	13º	5.01.11	ORG	Processos Organizacionais
13º	6.06.05	DH	Métricas de Produto	17º	6.05.13	LA	Manutenção
13º	6.06.06	DH	Métricas de Processo	18º	6.04.03	MP	Gerência de Riscos
19º	6.01.01	OC	Tipos de Software	18º	6.02	OC	Roteiros de Documentos
19º	6.04.13	MP	Manutenção	20º	6.05.10	LA	Codificação
19º	9.01.02	CL	Deficiências e Potenciais	20º	6.05.09	LA	Projeto (design)
19º	9.01.05	CL	Estrutura Organizacional	20º	6.05.07	LA	Engenharia de Sistemas
19º	0.01.01	LT	Conhecimento Teórico em ES	23º	6.04.08	MP	Análise de Requisitos
19º	0.01.02	LT	Inovações Tecnológicas	23º	6.03	OC	Itens de Software
19º	0.01.04	LT	LA da Indústria de Software	23º	5.01.05	ORG	Áreas Críticas para Objetivos

Tabela IV.45 – Tipos de Conhecimento quanto à Importância atribuída por Diretores considerando Projetos Grandes

Diretor considerando Projeto Grande (3 participantes)							
Mais Importantes				Menos Importantes			
1 ^o	6.01.02	OC	Processo de Software	1 ^o	7.01.08	FN	Relações Informais
1 ^o	6.01.03	OC	Avaliação do Processo	1 ^o	7.01.03	FN	Diretrizes e Normas
1 ^o	6.02	OC	Roteiros de Documentos	3 ^o	7.01.06	FN	Alocação dos Profissionais
1 ^o	6.03	OC	Itens de Software	3 ^o	7.01.05	FN	Estrutura Organizacional
1 ^o	6.04.04	MP	Gerência de Qualidade	3 ^o	7.01.02	FN	Deficiências e Potenciais
1 ^o	6.04.06	MP	Documentação	3 ^o	7.01.01	FN	Missão da Organização
1 ^o	6.04.10	MP	Codificação	7 ^o	9.01.08	CL	Relações Informais
1 ^o	6.04.15	MP	Reuso de Itens de Software	7 ^o	7.01.07	FN	Distribuição de Competências
1 ^o	6.04.16	MP	Melhoria de Processo	7 ^o	6.04.14	MP	Engenharia Reversa e Reeng
1 ^o	6.05.02	LA	Gerência de Projeto	10 ^o	9.01.07	CL	Distribuição de Competências
1 ^o	6.05.06	LA	Documentação	10 ^o	9.01.06	CL	Alocação dos Profissionais
1 ^o	6.05.08	LA	Análise de Requisitos	10 ^o	5.01.12	ORG	Contratos/Acordos
1 ^o	6.05.10	LA	Codificação	10 ^o	5.01.10	ORG	Relações Informais
1 ^o	6.05.13	LA	Manutenção	10 ^o	5.01.08	ORG	Alocação dos Profissionais
1 ^o	6.06.01	DH	Gerência de Projeto	15 ^o	9.01.05	CL	Estrutura Organizacional
1 ^o	6.06.03	DH	Gerência de Qualidade	15 ^o	9.01.01	CL	Missão da Organização
1 ^o	6.06.05	DH	Métricas de Produto	15 ^o	6.04.18	MP	Processo de Aquisição
1 ^o	6.06.06	DH	Métricas de Processo	15 ^o	5.01.07	ORG	Estrutura Organizacional
19 ^o	6.01.01	OC	Tipos de Software	15 ^o	5.01.06	ORG	Produtos e/ou Serviços
19 ^o	6.04.02	MP	Gerência de Projeto	20 ^o	9.01.03	CL	Diretrizes e Normas
19 ^o	6.04.07	MP	Engenharia de Sistemas	20 ^o	9.01.02	CL	Deficiências e Potenciais
19 ^o	6.04.08	MP	Análise de Requisitos	20 ^o	6.05.18	LA	Processo de Aquisição
19 ^o	6.04.11	MP	Testes	20 ^o	5.01.11	ORG	Processos Organizacionais
19 ^o	6.04.20	MP	Solução de Problemas	20 ^o	5.01.05	ORG	Áreas Críticas para Objetivos
25 ^o	5.01.02	ORG	Deficiências e Potenciais	25 ^o	5.01.09	ORG	Distribuição de Competências

ANEXO V

Processo de Configuração do Meta-ambiente TABA

As atividades que compõem o processo de configuração do Meta-ambiente TABA para uma organização são descritas a seguir, juntamente com as orientações para sua execução e os respectivos responsáveis.

➤ **Atividade 1: Contextualizar Configuração**

Esta atividade tem como objetivo contextualizar a configuração, indicando as características gerais da organização, a sua cultura na área de software e quais são os objetivos almejados com a configuração, de forma a fornecer os subsídios necessários à elaboração da proposta de configuração.

Responsáveis: Configurador de Ambientes e representantes da organização.

Sub-atividades:

Sub-atividade 1.1: Realizar Entrevistas

O objetivo é realizar as entrevistas necessárias para coletar as informações para caracterização da organização, identificação da cultura organizacional na área de software e identificação dos objetivos a serem atingidos a partir do Ambiente Configurado.

Sub-atividade 1.2: Caracterizar a Organização

O objetivo é registrar informações básicas sobre a organização que indiquem o seu tamanho e área de atuação, o que implica em fornecer respostas para as seguintes perguntas:

- Qual o tamanho da organização? (Micro – 1 a 9 pessoas, Pequena – 10 a 49 pessoas, Média – 50 a 99 pessoas, Grande – mais de 100 pessoas¹)
- A organização pertence a que tipo de indústria²? (Ex: Saúde, Telecomunicação, Transporte Aéreo e Software/Hardware)
- Quais são os domínios de conhecimento relacionados ao negócio da organização? (Ex: Cardiologia, Telefonia Celular, Aviação Comercial, Engenharia de Software e Redes de Computadores)
- Quais são os domínios de conhecimento envolvidos no apoio ao negócio da organização e para os quais sistemas de software têm sido ou pretendem ser desenvolvidos? (Ex: Administração Hospitalar, Contabilidade e Engenharia de Software)

¹ Esta classificação das organizações que desenvolvem e mantêm software foi proposta em [204], sendo também utilizada no capítulo 6.

² JONES [209] propõe, em estudos de software, o uso de códigos elaborados para a classificação de indústrias, listando os códigos das indústrias com maior população de desenvolvedores de software.

Sub-atividade 1.3: Identificar Cultura Organizacional na Área de Software

O objetivo é identificar os elementos que compõem a cultura da organização na área de software, o que envolve tipos de software produzidos, tamanho e complexidade dos projetos, tecnologias utilizadas, certificações e/ou avaliações obtidas e problemas enfrentados. Sendo assim, as respostas para as seguintes perguntas precisam ser registradas:

- Quantos profissionais da organização, aproximadamente, atuam na área de software?
- Quais são os tipos de software, considerando a finalidade, que são desenvolvidos pela organização? (Software para uso próprio, Pacote de software para comercialização, Software para uso por terceiros desenvolvido sob encomenda³)
- Quais são os tipos de software, considerando tecnologia envolvida, que são desenvolvidos pela organização? (Ex: Sistemas Web, Sistemas Cliente-Servidor e Sistemas Especialistas)
- Qual o tamanho e a complexidade dos projetos que têm sido conduzidos pela organização? (Quanto ao tamanho: Pequeno, Médio e Grande⁴, e quanto à complexidade: Baixa, Média, Alta e Software Crítico)
- Quais são os paradigmas de desenvolvimento utilizados na organização? (Estruturado, Orientado a Objeto, Baseado em Conhecimento)
- Quais são os métodos, técnicas e ferramentas utilizados na organização?
- A organização já possui processos de desenvolvimento e manutenção definidos? Há interesse em redefini-los?
- A organização possui certificação ISO relativa à área de software? Qual é o escopo?
- A organização possui avaliação segundo o CMM ou CMMI? Em que nível? Formal ou informal?
- Caso a organização não tenha processos definidos, quais são os documentos produzidos na organização em seus projetos de software? Existem procedimentos já definidos? Há a necessidade de executar atividades que não são geralmente executadas em outras organizações ou que são executadas de forma particular na organização?

³ As opções fornecidas são uma simplificação da classificação utilizada em [204].

⁴ Esta classificação dos projetos quanto ao tamanho obedece ao critério definido na tabela 6.6 apresentada no capítulo 6.

- Quais os principais problemas enfrentados pela organização em seus projetos de software? (Ex: Falta de conhecimento sobre o domínio, falta de experiência dos desenvolvedores, alta rotatividade, estimativas não realistas e muito re-trabalho)

Sub-atividade 1.4: Identificar Objetivos para a Configuração do Ambiente

O objetivo é identificar os objetivos da organização na área de software que estão motivando a configuração do ambiente. Desta forma, respostas para as seguintes perguntas precisam ser fornecidas:

- Quais são os objetivos da organização na área de software? (Ex: Obter certificação ISO, atingir um determinado nível de maturidade segundo o CMM ou CMMI, adotar uma nova tecnologia, atuar em novo domínio de conhecimento e tratar um dos problemas freqüentes nos projetos da organização)
- Quais desses objetivos devem ser apoiados pelo ambiente a ser configurado?

➤ **Atividade 2: Elaborar Proposta para Configuração do Ambiente**

Esta atividade tem como objetivo elaborar a proposta de configuração de um ambiente para a organização, de acordo com as informações previamente obtidas, o que implica em especificar o conteúdo e as ferramentas que serão disponibilizadas no Ambiente Configurado e em elaborar planos de organização e custos, além de cronograma para a configuração.

Responsável: Configurador de Ambientes.

Sub-atividades:

Sub-atividade 2.1: Identificar Teorias de Domínio

O objetivo é identificar, considerando os domínios de conhecimento mencionados nas atividades anteriores, as Teorias do Domínio que serão disponibilizadas no Ambiente Configurado, definindo os seus respectivos propósitos, o que implica em definir qual será a sua utilização e delimitar o seu escopo. Esta atividade envolve, portanto, uma descrição geral do propósito de cada Teoria do Domínio, a elaboração de cenários de motivação que demonstrem a sua utilização e a formulação de questões gerais de competência para delimitação do seu escopo, que são questões abrangentes que indicam aspectos a serem tratados pela mesma. Em caso de uma Teoria do Domínio já ter sido definida anteriormente, identifica-se a necessidade ou não de evoluí-la.

Sub-atividade 2.2: Identificar Processos

O objetivo é identificar os novos processos a serem definidos e os processos já existentes que precisam ser revistos, de forma a fornecer para a organização um processo padrão contemplando os processos de ciclo de vida de desenvolvimento e manutenção, além dos processos especializados para cada contexto de desenvolvimento de software existente na organização. O processo padrão é a base para a definição dos processos especializados, que o adaptam de acordo com os paradigmas de desenvolvimento, métodos e/ou tipos de software que compõem os contextos específicos de desenvolvimento de software na organização.

Sub-atividade 2.3: Identificar Ferramentas

O objetivo é identificar as ferramentas que serão disponibilizadas para uso no Ambiente Configurado e/ou nos Ambientes Instanciados a partir dele, o que inclui tanto ferramentas internas quanto ferramentas externas. Nesta atividade, deve ser verificada a necessidade de construção e/ou aquisição de novas ferramentas. As atividades necessárias à aquisição de uma ferramenta constituem um processo independente do processo de configuração, devendo ser executadas de acordo com a norma internacional ISO/IEC 14598-5 Parte 4 [212] e o processo para aquisição de produtos de software da organização. Já as atividades necessárias à construção de uma ferramenta devem ser apoiadas por um ambiente instanciado com esta finalidade, adotando-se o processo de construção de ferramentas definido para o projeto TABA.

Sub-atividade 2.4: Definir Equipes e Responsabilidades

O objetivo é estabelecer quem será o coordenador do projeto de configuração do Meta-ambiente TABA para a organização e definir equipes responsáveis pelas macro-atividades que fazem parte da proposta de configuração. Exemplos de macro-atividades são: Definição dos Processos, Definição da Teoria do Domínio, Construção de Ferramentas, Treinamento e Acompanhamento de Projeto Piloto.

Sub-atividade 2.5: Definir Cronograma

O objetivo é estabelecer uma cronograma para a execução das macro-atividades que compõem a proposta de configuração.

Sub-atividade 2.6: Definir Custos

O objetivo é elaborar uma planilha de custo por macro-atividades, estabelecendo também o custo total da configuração do Meta-ambiente TABA para a organização.

Sub-atividade 2.7: Enviar Proposta

O objetivo é enviar a proposta elaborada ao longo desta atividade para que os Representantes da Organização a analisem, identifiquem possíveis dúvidas e problemas, podendo, assim, fornecer um parecer sobre a mesma.

➤ **Atividade 3: Registrar Parecer sobre a Proposta**

Esta atividade tem como objetivo registrar o parecer dos Representantes da Organização sobre a proposta de configuração, indicando a sua aprovação, a necessidade de modificações e ajustes, ou a não aprovação da proposta. No caso da necessidade de modificações e ajustes, uma data para envio da nova proposta é estabelecida.

Responsável: Configurador de Ambientes.

➤ **Atividade 4: Definir Processo Padrão**

Esta atividade tem como objetivo definir o processo padrão da organização, contemplando os processos de desenvolvimento e de manutenção de software.

Responsável: Configurador de Ambientes.

Sub-atividades:

Sub-atividade 4.1: Caracterizar Processo Padrão

O objetivo é caracterizar o processo padrão a ser definido, estabelecendo quais são os processos de ciclo de vida que devem compor o processo padrão e se as definições dos processos de ciclo de vida devem ser baseadas ou não em processos já definidos anteriormente. Caso não devam ser baseados em processos já definidos, a norma ISO/IEC 12207 [140] é utilizada com padrão de referência.

Sub-atividade 4.2: Incluir Atividades do Processo Padrão Base (quando pertinente)

O objetivo é incluir, automaticamente, no novo processo padrão da organização, as atividades do processo padrão selecionado como base para a definição do processo. Se necessário, atividades podem ser alteradas ou excluídas.

Sub-atividade 4.3: Incluir Atividades ISO/IEC 12207

O objetivo é selecionar as atividades da norma ISO/IEC 12207 [140] a serem incluídas no processo padrão da organização, o que é de fundamental importância quando um requisito do processo padrão é estar de acordo com a norma.

Sub-atividade 4.4: Incluir Atividades do Tipo de Software

O objetivo é selecionar, para inclusão no processo padrão da organização, as atividades que são próprias do desenvolvimento ou da manutenção de determinados tipos de software. Atividades desta natureza só devem ser incluídas no processo padrão de uma organização quando são aplicáveis a qualquer projeto de software conduzido na organização. Um exemplo é quando a organização só desenvolve e mantém software crítico de tempo real. Nesta situação, tanto atividades próprias de software crítico quanto atividades próprias de software de tempo real podem e devem ser incluídas no processo padrão.

Sub-atividade 4.5: Incluir Atividades Próprias da Organização

O objetivo é definir e incluir as atividades que devem fazer parte do processo padrão e que são específicas da organização para a qual o ambiente está sendo configurado. Uma exemplo de atividade própria da organização pode ser a que estabelece a elaboração de uma proposta comercial.

Sub-atividade 4.6: Incluir Atividades Orientadas a Domínio

O objetivo é incluir as atividades orientadas a domínio no processo padrão da organização. No Meta-ambiente TABA, tem-se cadastrado a atividade de Investigação do Domínio, a ser incluída como sub-atividade de outras atividades, estabelecendo a pesquisa e o uso do conhecimento sobre o domínio que sejam necessários para a execução da atividade correspondente [20]. No entanto, atividades orientadas a domínio mais específicas podem ser cadastradas para fornecer aos desenvolvedores descrições mais voltadas para as atividades a serem apoiadas, se assim for desejado. Atividades orientadas a domínio só devem ser incluídas no processo padrão se estiver prevista, na proposta de configuração, a inclusão de pelo menos uma Teoria de Domínio referente a um domínio de negócio.

Sub-atividade 4.7: Estabelecer Atividades Obrigatórias

O objetivo é estabelecer quais atividades do processo padrão são obrigatórias e, conseqüentemente, não podem ser retiradas dos processos que serão posteriormente definidos a partir do mesmo. As atividades obrigatórias compõem o conjunto mínimo de atividades a serem executadas independente das características específicas de cada projeto.

Sub-atividade 4.8: Adaptar Roteiros de Documentos para a Organização

O objetivo é adaptar roteiros de documentos disponíveis na Meta-ambiente TABA para a organização, de forma a disponibilizá-los para uso no Ambiente Configurado.

➤ **Atividade 5: Definir Processo Especializado**

Esta atividade tem como objetivo definir, a partir do processo padrão, processos especializados para os paradigmas de desenvolvimento utilizados na organização e, opcionalmente, para diferentes tipos de software.

Responsável: Configurador de Ambientes.

Sub-atividades:

Sub-atividade 5.1: Especializar Atividades do Processo Padrão

O objetivo é alterar a descrição das atividades previstas no processo sendo especializado, de forma a torná-las mais adequadas ao paradigma e respectivos métodos de desenvolvimento. Isto inclui a possível inclusão ou especialização de sub-atividades, definição de novos produtos e utilização de novas ferramentas.

Sub-atividade 5.2: Incluir Atividades do Tipo de Software

O objetivo é incluir, em um processo especializado, as atividades que são próprias do desenvolvimento ou da manutenção de determinados tipos de software, dando origem a um processo especializado tanto para o paradigma quanto para um ou mais tipos de software. No entanto, nem sempre a execução desta atividade é necessária, pois as atividades dessa natureza podem já ter sido incluídas no processo padrão (sub-atividade 4.4: *Incluir Atividades do Tipo de Software*), podem ter a inclusão postergada até a instanciação do processo para um projeto específico, ou os tipos de software desenvolvidos na organização podem não requerer a execução de atividades específicas.

Sub-atividade 5.3: Estabelecer Atividades Obrigatórias

O objetivo é determinar quais atividades do processo especializado são obrigatórias e, conseqüentemente, não podem ser retiradas durante a definição dos processos para os projetos específicos. As atividades definidas como obrigatórias no processo padrão permanecem como obrigatórias.

Sub-atividade 5.4: Especializar Roteiros de Documentos

O objetivo é especializar, quando necessário e de acordo com as características de cada processo especializado, os roteiros de documentos definidos para a organização, de forma a poder disponibilizar estes roteiros para uso no Ambiente Configurado.

➤ **Atividade 6: Definir Teoria do Domínio**

Esta atividade tem como objetivo definir uma Teoria do Domínio para um determinado domínio de conhecimento.

Responsáveis: Engenheiro de Conhecimento e especialistas no domínio.

Sub-atividades:

Sub-atividade 6.1: Revisar Propósito

O objetivo é revisar o propósito estabelecido para a Teoria do Domínio na sub-atividade 2.1 (*Identificar Teorias de Domínio*), efetuando os ajustes necessários na descrição geral do propósito, nos cenários de motivação e nas questões gerais de competência.

Sub-atividade 6.2: Especificar Requisitos

O objetivo é especificar os requisitos da Teoria do Domínio através da formulação de questões de competência, que são as questões que a Teoria do Domínio deve possibilitar que sejam respondidas.

Sub-atividade 6.3: Capturar Teoria do Domínio

O objetivo é identificar conceitos, relações, propriedades e restrições da ontologia do domínio que compõe a Teoria do Domínio. Parte-se de um conjunto preliminar de termos e de frases potencialmente relevantes, que são agrupados de acordo com a relação semântica existente, dando origem a áreas de trabalho. Para cada área de trabalho, são geradas as definições preliminares dos conceitos representados pelos termos, substituídos os termos ambíguos, eliminados os conceitos redundantes e solucionadas as inconsistências. A partir das frases potencialmente relevantes, são identificadas as relações entre os conceitos. Em seguida, são identificadas as propriedades necessárias nos conceitos e relações, de forma que as questões de competência sejam respondidas. As áreas de trabalho dão, naturalmente, origem às Subteorias do Domínio. A identificação das Subteorias de um Domínio permite verificar a possibilidade de integração de Subteorias já existentes ou de parte de seus conceitos.

Sub-atividade 6.4 Conceituar Teoria do Domínio

O objetivo é gerar definições precisas e não ambíguas para a ontologia capturada. Nesta sub-atividade devem ser elaborados dicionários de conceitos, de propriedades e de relações entre conceitos, árvores de classificação de conceitos (quando pertinentes) e, também quando pertinente, uma tabela de fórmulas utilizadas para inferir valores numéricos que relacionam propriedades. Para concluir a atividade, restrições axiomáticas (axiomas que estabelecem restrições para as relações entre conceitos ou entre propriedades de um conceito) devem ser descritas em linguagem natural. Estas restrições axiomáticas são definidas com base nas questões de competência.

Sub-atividade 6.5: Formalizar Teoria do Domínio

O objetivo é descrever a ontologia do domínio utilizando uma linguagem de representação que elimine as possíveis ambigüidades provenientes do uso da linguagem natural. A depender da linguagem utilizada, todos os axiomas tem que ser explicitamente definidos, inclusive os axiomas que representam especialização e composição e os que especificam domínio de valores para os atributos.

Sub-atividade 6.6: Identificar Tarefas

O objetivo é identificar quais são as tarefas que são executadas no contexto definido por cada Subteoria do Domínio. Quando uma tarefa identificada não tiver sido ainda descrita no Meta-ambiente TABA, a atividade 7 (*Descrever Tarefa*) deve ser executada.

Sub-atividade 6.7: Avaliar Teoria do Domínio

O objetivo é avaliar a Teoria de Domínio definida, de forma a assegurar que ela cumpre o seu propósito, satisfazendo os requisitos estabelecidos pelas questões de competência e realmente sendo capaz de descrever o mundo real para o qual foi projetada. Além disso, também é objetivo desta atividade avaliar a Teoria do Domínio segundo os critérios de qualidade apresentados em [20]. Desta forma, esta atividade compreende tanto a verificação quanto validação da Teoria do Domínio.

Sub-atividade 6.8: Implementar Teoria do Domínio

O objetivo é implementar a formalização da Teoria do Domínio no Meta-ambiente. Com isso, a Teoria do Domínio passa a estar disponível como módulos de conhecimento para que possa ser instanciada e acessada. A integração das Subteorias do Domínio já disponíveis no meta-ambiente é fundamental para manter a completude e a consistência da Teoria do Domínio.

As sub-atividades 6.2, 6.3 e 6.4 (*Especificar Requisitos*, *Capturar Teoria do Domínio* e *Conceituar Teoria do Domínio*) são executadas de forma iterativa. A primeira iteração tem como objetivo formular as questões de competência, obter um conjunto preliminar de termos e dividi-los em áreas de trabalho. Tantas iterações quanto necessárias devem ser realizadas a fim de obter o modelo parcial da Teoria do Domínio, o que não inclui a descrição das restrições axiomáticas em linguagem natural. Um dos possíveis critérios para definição do número de iterações é o número de áreas de trabalho. A última iteração tem como objetivo completar o modelo da Teoria do Domínio com as restrições axiomáticas.

➤ **Atividade 7: Descrever Tarefa**

Esta atividade tem como objetivo descrever uma tarefa genérica que tenha sido identificada como pertinente ao contexto de uma Subteoria do Domínio. Apesar da necessidade de descrição da tarefa estar associada à definição de uma Subteoria, a descrição da tarefa em si é independente do domínio de conhecimento específico, permitindo a formação de uma biblioteca de tarefas que pode ser utilizada para descrever o conhecimento de diversos domínios. Novas tarefas são descritas e acrescentadas a biblioteca na medida em que são necessárias.

Responsáveis: Engenheiro de Conhecimento e especialistas na execução da tarefa.

Sub-atividades:

Sub-atividade 7.1: Descrever Tarefa em Linguagem Natural

O objetivo é descrever a tarefa em linguagem natural, o que implica em descrever o objetivo da tarefa e os elementos (objetos e ações) que são necessários para a solução do problema proposto pela tarefa.

Sub-atividade 7.2: Definir Estratégia para Solução da Tarefa

O objetivo é escolher um método para a solução do problema proposto pela tarefa e, a depender da estratégia adotada pelo método escolhido, descrever a solução do problema em linguagem natural. A estratégia adotada pelo método determina se a tarefa deve ser considerada elementar ou composta, ou seja, se o problema proposto pela tarefa deve ser resolvido diretamente através de inferências ou se é necessário decompor a tarefa original em tarefas mais simples (sub-tarefas). Quando a tarefa for considerada composta, a solução do problema deve ser descrita em linguagem natural, especificando as suas sub-tarefas e o fluxo de controle necessário.

Sub-atividade 7.3: Capturar Conceitos e Relações da Tarefa

O objetivo é identificar, a partir dos objetos mencionados na descrição em linguagem natural da tarefa, a lista de conceitos envolvidos na solução do problema e as relações entre estes conceitos. Os conceitos e relações compõem a ontologia da tarefa, que permite simular o processo de solução do problema, mas não demonstra com o problema pode ser resolvido. Os conceitos da tarefa são, na realidade, os papéis de conhecimento que serão preenchidos por conceitos do domínio quando a tarefa estiver sendo executada no contexto de um domínio específico.

Sub-atividade 7.4: Descrever Fluxo de Controle da Tarefa

O objetivo é descrever, em linguagem estruturada, um procedimento que represente o fluxo de controle necessário para a solução do problema quando a tarefa tiver sido considerada composta. Isto implica que o procedimento deve especificar a ordem de execução das sub-tarefas que compõem a tarefa.

Sub-atividade 7.5: Formalizar Descrição da Tarefa

O objetivo é obter uma representação formal da descrição da tarefa. Tanto os conceitos, relações e restrições da ontologia da tarefa quanto o fluxo de controle anteriormente definidos precisam ser descritos em uma linguagem formal, eliminando ambigüidades. No entanto, se a tarefa tiver sido considerada elementar, ao invés da formalização do fluxo de controle, são definidas regras de inferência, também em linguagem formal, que determinam como solucionar a tarefa a partir da interação entre os seus conceitos.

Sub-atividade 7.6: Avaliar Descrição da Tarefa

O objetivo é avaliar a Descrição da Tarefa, de forma a assegurar que ela realmente descreve o problema proposto pela tarefa e apresenta uma solução para o problema que seja válida no mundo real. Se a tarefa for composta, suas sub-tarefas devem ter sido avaliadas previamente. Outro objetivo desta atividade é avaliar a Descrição da Tarefa segundo um subconjunto de critérios de qualidade apresentados em [20] que também se aplicam à avaliação de Descrições de Tarefa. Desta forma, esta atividade compreende tanto a verificação quanto validação da Descrição da Tarefa.

Sub-atividade 7.7: Implementar Descrição da Tarefa

O objetivo é implementar a Descrição da Tarefa no meta-ambiente, codificando a sua formalização e, quando a tarefa for composta, integrando a sua descrição com as descrições de suas sub-tarefas. Como isso, a Descrição da Tarefa torna-se disponível para uso e assegura-se a sua completude e consistência.

No caso da estratégia adotada pelo método (sub-atividade 7.2: *Definir Estratégia para Solução da Tarefa*) considerar a tarefa como composta, todas as sub-atividades acima são executadas de forma recursiva para cada sub-tarefa na qual a tarefa tenha sido decomposta, a menos que a sub-tarefa já conste na biblioteca de tarefas. Conforme estabelecido por ZLOT [137], a descrição das sub-tarefas acrescenta novos conceitos aos já identificados para a tarefa, de forma a permitir a comunicação entre duas ou mais sub-tarefas.

➤ **Atividade 8: Incluir Novas Ferramentas nas Descrições dos Processos**

Ferramentas que tenham sido construídas após a definição dos processos padrão e especializados precisam ser incluídas na descrição das atividades que as utilizam, para que, no momento da configuração, possam ser integradas ao Ambiente Configurado. O objetivo desta atividade é, então, introduzir a referência a estas ferramentas na descrição dos processos padrão e/ou especializados.

Responsável: Configurador de Ambientes.

➤ **Atividade 9: Criar Ambiente Configurado**

O objetivo desta atividade é criar o Ambiente Configurado para a organização. Esta atividade conclui o processo de configuração.

Responsável: Configurador de Ambientes.

Sub-atividades:

Sub-atividade 9.1: Definir Ambiente Configurado

O objetivo é definir o Ambiente Configurado, selecionando o processo padrão e as Teorias de Domínio a serem fornecidas como conteúdo inicial do ambiente.

Sub-atividade 9.2: Gerar Ambiente Configurado

O objetivo é gerar o código executável do Ambiente Configurado, opcionalmente disponibilizando o código fonte, a partir de informações como logomarca da organização, ícone para acesso e diretórios de armazenamento.

Sub-atividade 9.3: Testar Ambiente Configurado

O objetivo é testar o Ambiente Configurado para assegurar que ele foi gerado corretamente.

ANEXO VI

Atividades de Preparação
do Ambiente Configurado

Uma vez que o Ambiente Configurado é entregue à organização, o seu Gerente de Conhecimento prepara o ambiente para utilização, introduzindo conhecimento sobre a organização, além de itens de conhecimento sobre desenvolvimento e manutenção de software que tenham sido anteriormente adquiridos e estejam prontos para serem disponibilizados no ambiente.

➤ **Atividade: Descrever Organização**

Esta atividade tem como objetivo obter e registrar informações sobre a organização, de forma que estas informações possam ser disponibilizadas para apoiar os seus vários projetos de software. A descrição da organização deve ser feita de forma evolutiva, começando pelas informações com uso previsto a curto prazo ou que foram consideradas mais importantes pelos desenvolvedores de software da organização, sempre considerando a relação custo-benefício.

Responsáveis: Gerente de Conhecimento e especialistas das áreas envolvidas.

Sub-atividades:

Sub-atividade: Descrever Aspectos Gerais

O objetivo é obter informações que definem como a organização interage com o ambiente externo, registrando-as no Ambiente Configurado. Desta forma, pode ser relevante obter respostas para as seguintes perguntas:

- Qual é a missão da organização?
- Quais são os seus objetivos? Existe data prevista para que estes objetivos sejam atingidos?
- Quais são os serviços e/ou artefatos oferecidos? Quais são as empresas clientes destes serviços e/ou artefatos?
- Que projetos existem e que organizações estão envolvidas?

Sub-atividade: Descrever Estrutura Organizacional

O objetivo é obter informações sobre como a organização se encontra estruturada, registrando estas informações no Ambiente Configurado. Desta forma, pode ser relevante obter respostas para as seguintes perguntas:

- Quais são as unidades organizacionais existentes e como elas se relacionam?
- Quais são as posições existentes nas unidades organizacionais e como elas se relacionam? Que competências são requeridas? Quais são as pessoas que estão alocadas e que competências possuem?
- Existem comissões na organização? Com que propósito? Quem são as pessoas que fazem parte destas comissões?

Sub-atividade: Descrever Processos Organizacionais

O objetivo é obter informações sobre os processos executados na organização, representando-os graficamente no Ambiente Configurado. Desta forma, pode ser relevante obter respostas para as seguintes perguntas:

- Estes processos da organização obedecem a alguma norma? Qual?
- Quais são as atividades que compõem os processos organizacionais e como elas se relacionam?
- Quais são os insumos e produtos de cada atividade e que competências e recursos materiais e humanos são requeridos para a sua execução?
- Existe algum procedimento que indique como uma determinada atividade deve ser executada?

➤ **Atividade: Descrever Organização Cliente**

Esta atividade só deve ser executada se a organização tem como atividade de negócio o desenvolvimento e manutenção de software. O objetivo é obter e registrar informações sobre uma organização cliente, de forma que estas informações possam ser disponibilizadas para apoiar os vários projetos de software que envolvem esta organização. A descrição da organização cliente deve ser feita de forma evolutiva, considerando sempre a relação custo-benefício. O foco da atividade são as informações relevantes para os projetos de software da organização e não a obtenção de uma descrição completa da organização cliente. Além disso, as informações registradas no Ambiente Configurado sobre uma organização cliente evoluem a cada novo projeto envolvendo esta organização.

Responsáveis: Gerente de Conhecimento e especialistas em projetos com o cliente.

Sub-atividades:

Sub-atividade: Descrever Aspectos Gerais

O objetivo é obter informações que definem como a organização cliente interage com o ambiente externo, registrando-as no Ambiente Configurado.

Sub-atividade: Descrever Estrutura Organizacional

O objetivo é obter informações sobre como a organização cliente se encontra estruturada, registrando estas informações no Ambiente Configurado.

Sub-atividade: Descrever Processos Organizacionais

O objetivo é obter informações sobre os processos executados na organização cliente, representando-os graficamente no Ambiente Configurado.

As perguntas a serem respondidas em cada uma das sub-atividades acima podem ser extraídas das perguntas mencionadas nas sub-atividades de mesmo nome executadas para a própria organização.

➤ **Adquirir Conhecimento Inicial**

Esta atividade tem como objetivo disponibilizar, no Ambiente Configurado, itens de conhecimento sobre desenvolvimento e manutenção de software que já estavam disponíveis para a organização antes da configuração em algum formato, local e meio de armazenamento. Esta atividade deve ser executada segundo o processo de captura de conhecimento proposto por MONTONI [31], de forma a possibilitar a filtragem de itens de conhecimento realmente relevantes, assegurar que a representação está adequada para a sua reutilização e que o conteúdo está claro e pode ser facilmente compreendido.

Responsáveis: Gerente de Conhecimento, Comitê de Avaliação e antigos detentores dos conhecimentos.

ANEXO VII

Visão Parcial do Modelo de Classes da Estação TABA

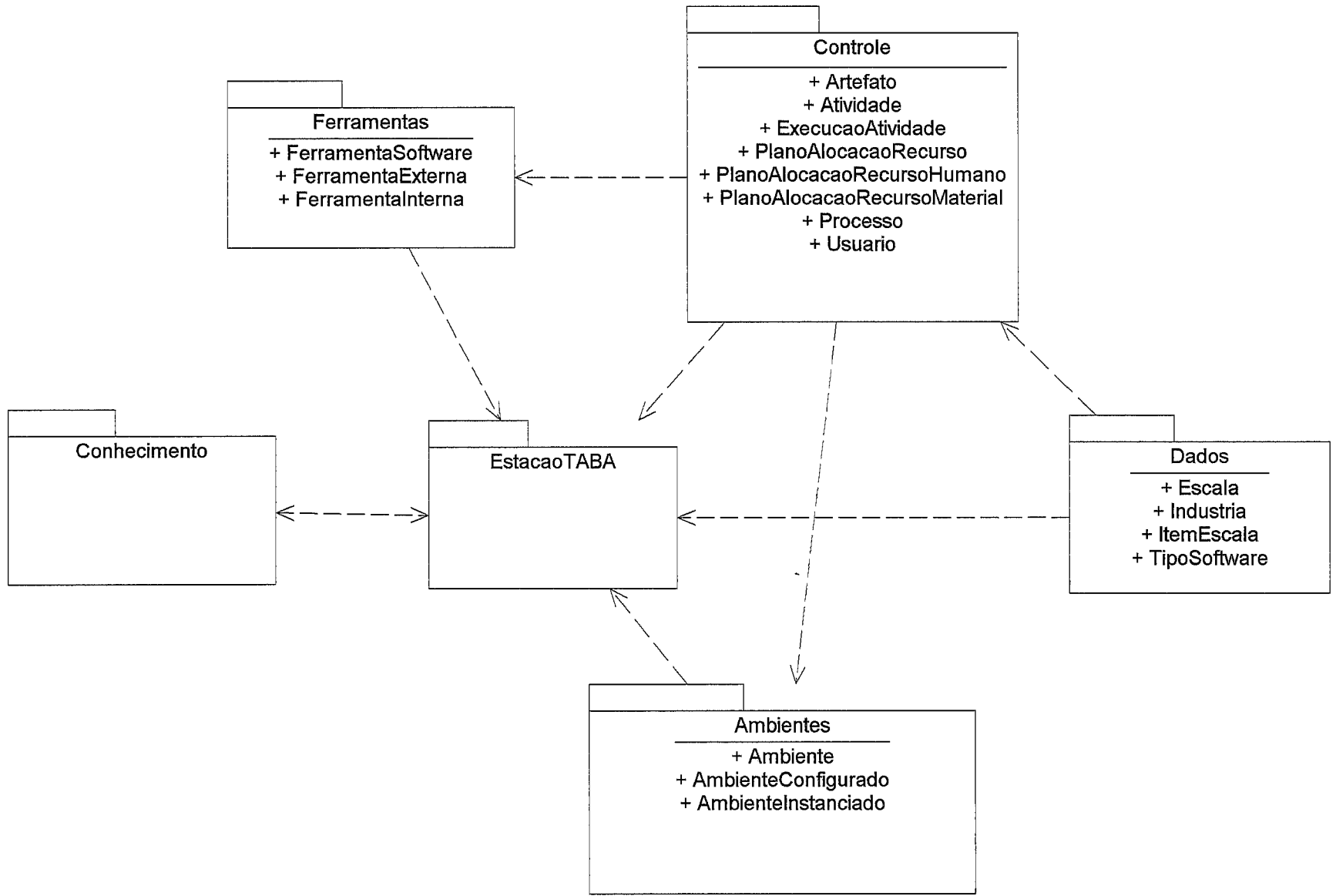


Figura VII.1 - Modelo de Pacotes da Estação TABA

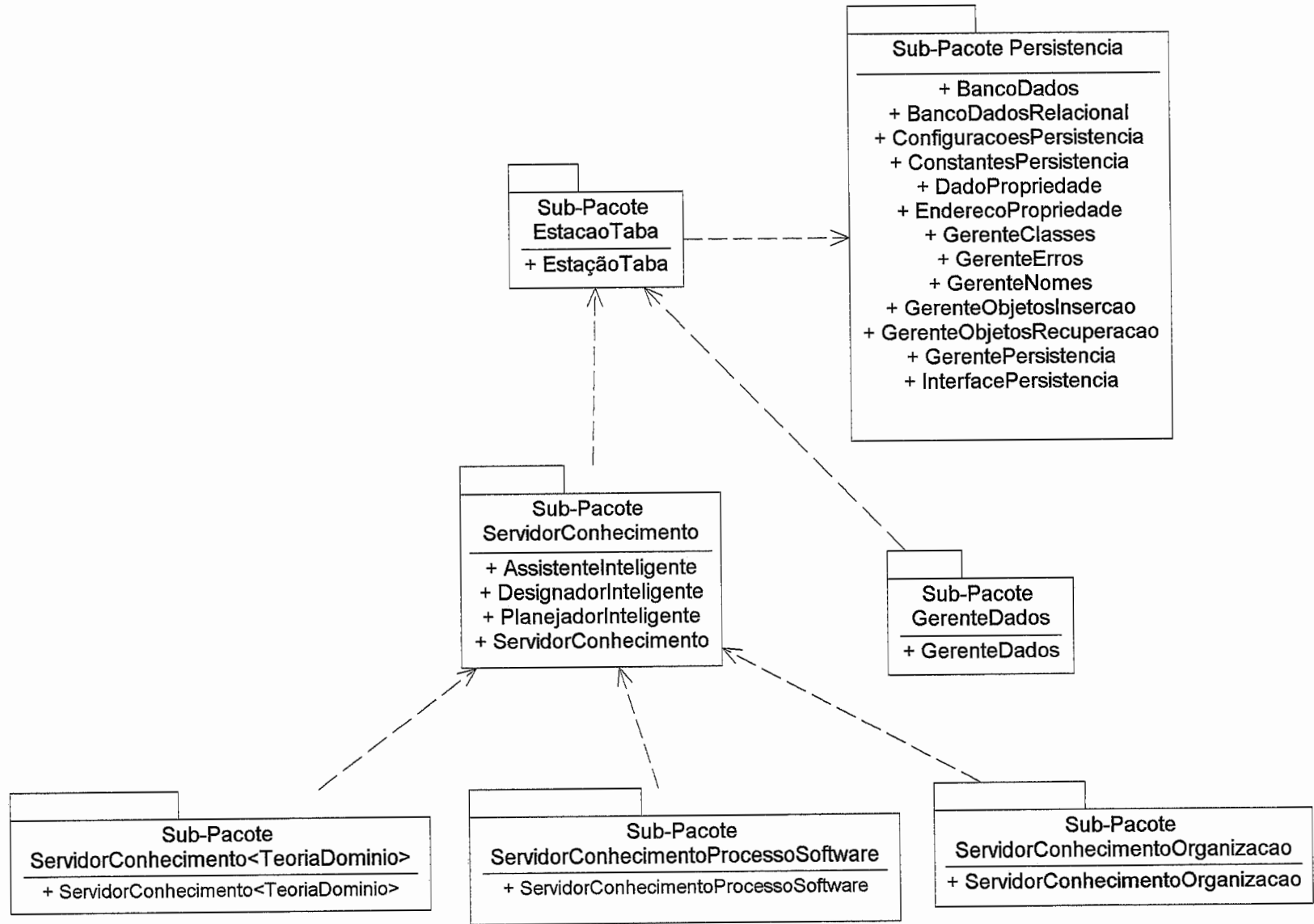


Figura VII.1.1 - Modelo de Sub-Pacotes do Pacote Estação TABA

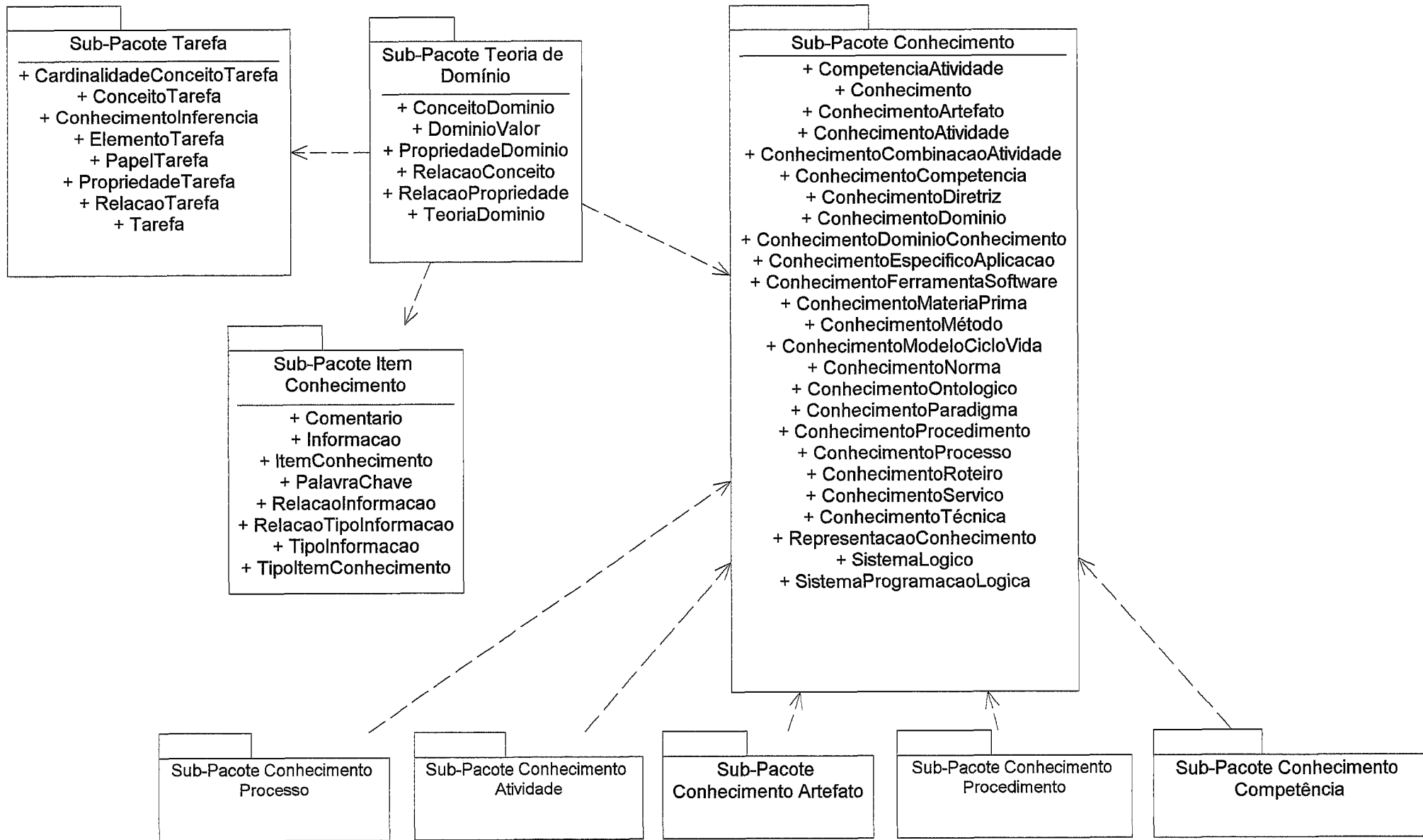


Figura VII.1.2 - Modelo de Sub-Pacotes do Pacote Conhecimento

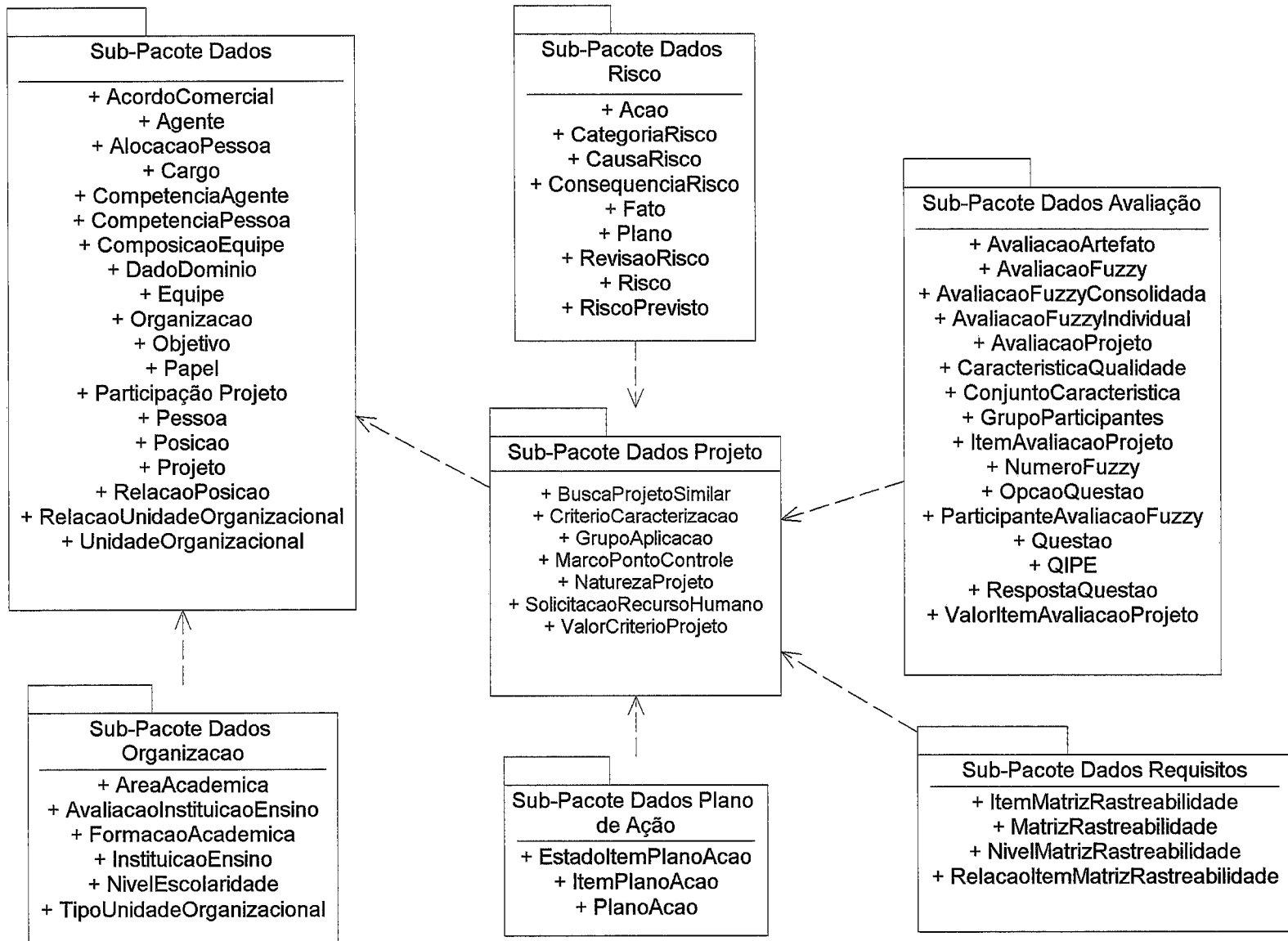
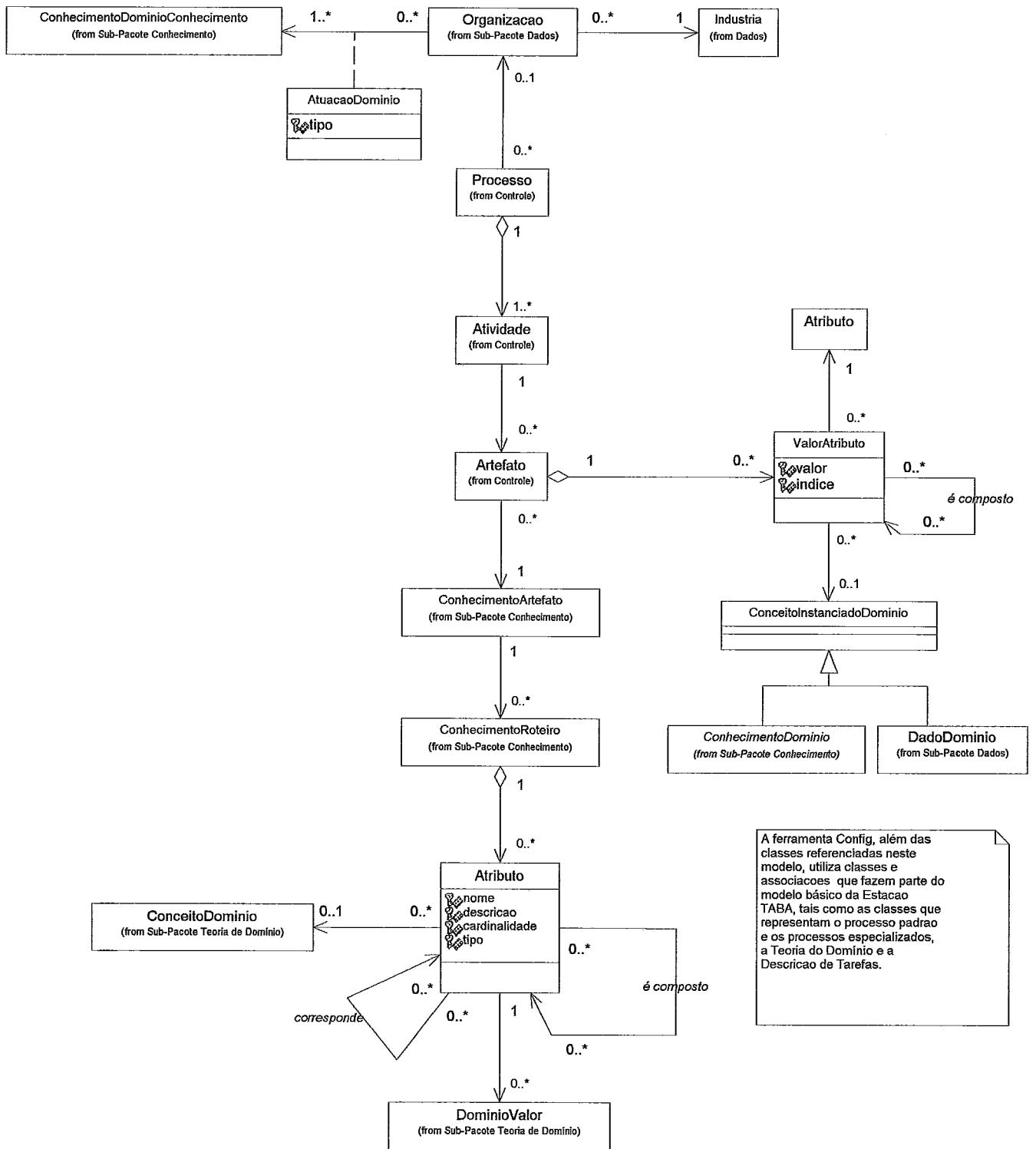


Figura VII.1.3 - Modelo de Sub-Pacotes do Pacote Dados



A ferramenta Config, além das classes referenciadas neste modelo, utiliza classes e associações que fazem parte do modelo básico da Estação TABA, tais como as classes que representam o processo padrão e os processos especializados, a Teoria do Dominio e a Descrição de Tarefas.

Figura VII.2 - Modelo de Classes da Ferramenta Config

ANEXO VIII

Descrição das Ferramentas Config e Procknow

VIII.1 Descrição da Ferramenta Config

A figura VIII.1 mostra a tela inicial do *Config* com a primeira atividade do processo de configuração a ser executada, *Realizar Entrevistas*. Esta atividade é uma sub-atividade da atividade *Contextualizar Configuração*.

Na tela *Realizar Entrevistas* (figura VIII.1), o configurador de ambientes informa a organização para a qual o Ambiente Configurado será criado e agenda as entrevistas necessárias à contextualização da configuração. Para cada entrevista, o configurador pode especificar os prováveis participantes e um roteiro a ser seguido. A ferramenta automaticamente sugere um roteiro com todas as perguntas referentes à contextualização da configuração, que o configurador manipula como quiser, compondo o roteiro da entrevista. Quando uma entrevista é realizada, os participantes são confirmados, sendo também registrados o resultado e possíveis aspectos a esclarecer. No entanto, as principais informações coletadas podem e devem ser diretamente registradas nas telas das figuras VIII.2 a VIII.6, que tratam da caracterização da organização, da identificação da cultura organizacional na área de software e da identificação dos objetivos estabelecidos para configuração do ambiente.

The screenshot shows the 'Configuração de Ambiente' window. On the left is a sidebar with a tree view under 'Contextualizar Configuração'. The main area is titled 'Realizar Entrevistas' and contains the following elements:

- Organização:** A dropdown menu with the selected value 'Unidade de Cardiologia e Cirurgia Cardiovascular / Fundação Bahiana de Cardiologia'.
- Entrevistas:** A table with columns 'Identificação', 'Data/Hora', and 'Situação'.

Identificação	Data/Hora	Situação
Entrevista com Patrocinador	12/08/2003	Agendada
Entrevista com Responsável Técnico	15/08/2003	Agendada
- Participantes:** A table with columns 'Nome', 'Telefone', and 'E-mail'.

Nome	Telefone	E-mail
Alvaro Rabelo		
- Roteiro:** A text area with the label 'Objetivos da Configuração' and a scroll bar.
- Resultado:** A text area with a scroll bar.
- Aspectos a Esclarecer:** A text area with a scroll bar.

Figura VIII.1 – Tela Inicial do Config: Realização de Entrevistas

Na tela *Caracterizar a Organização* (figura VIII.2), o configurador de ambientes descreve brevemente a organização e seleciona as opções fornecidas pela ferramenta que respondem corretamente às perguntas sobre tamanho da organização, indústria a que ela pertence, e domínios de conhecimento do negócio e de apoio ao negócio nos quais a organização devolve ou tem interesse de desenvolver sistemas de *software*.

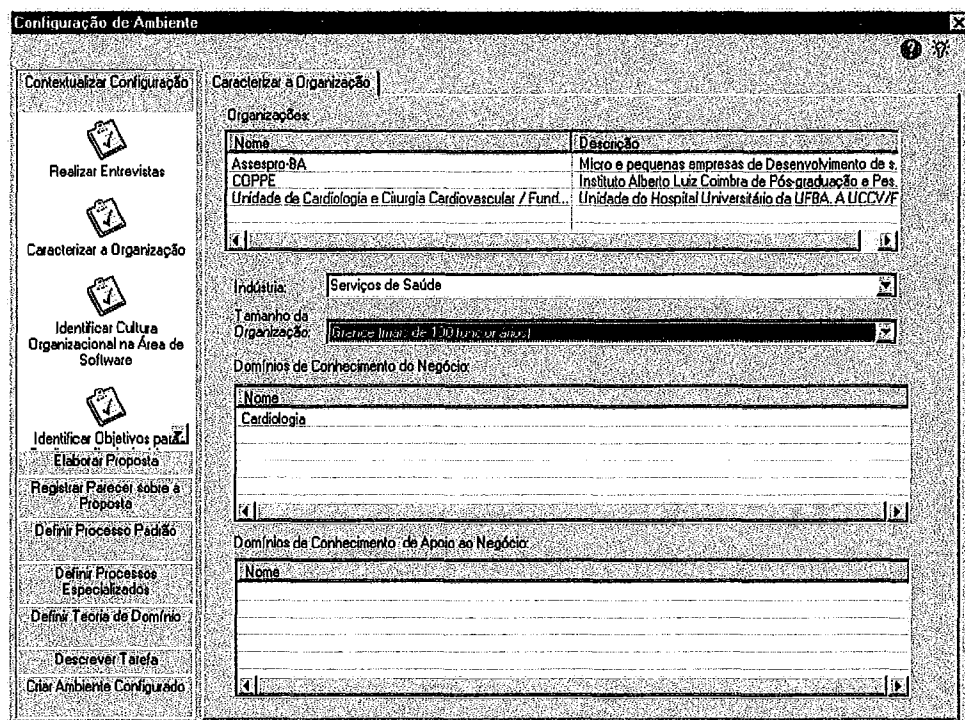


Figura VIII.2 – Tela de Caracterização da Organização

As telas exibidas nas figuras VIII.3, VIII.4 e VIII.5 organizam as informações referentes à atividade de *Identificar Cultura Organizacional na Área de Software*. No que se refere às informações gerais (figura VIII.3), o configurador informa o número de profissionais na área de software, as certificações e/ou avaliações obtidas, os tipos de software desenvolvidos quanto à finalidade e à tecnologia, o tamanho e complexidade dos projetos, além dos principais problemas enfrentados. No que se refere a processos (figura VIII.4), o configurador informa os processos, roteiros e procedimentos porventura já previamente definidos para a organização e, quando não houver processos definidos, são informadas as atividades comumente executadas na organização. Quando o configurador pressiona o botão com a reticências ao lado do campo *Situação*, a ferramenta abre uma janela que permite anexar o respectivo documento. Por fim, no que se refere à tecnologia (figura VIII.5), o configurador informa os paradigmas de desenvolvimento, métodos, técnicas e ferramentas já adotados na organização.

Configuração de Ambiente

Identificar Cultura Organizacional na Área de Software

Informações Gerais | Processos | Tecnologias

Número de Profissionais na Área de Software: 13

Certificações/Avaliações:

Nome	Tipo	Descrição
ISO 9000:2000	Formal	

Tipos de Software quanto à Finalidade:

Nome
Software para uso próprio

Tipos de Software quanto à Tecnologia:

Nome
Sistema para Web

Tamanho dos Projetos:

Pequeno (<14 homem-mês)

Médio (<36,5 homem-mês)

Grande (>= 36,5 homem-mês)

Complexidade dos Projetos:

Baixa

Média

Alta

Software de Segt

Principais Problemas em Projetos de Software:

Figura VIII.3 – Tela de Identificação da Cultura Organizacional na Área de Software: Seção de Informações Gerais

Configuração de Ambiente

Identificar Cultura Organizacional na Área de Software

Informações Gerais | Processos | Tecnologias

Processos Previamente Definidos:

Nome	Situação
Processo Padrão de Desenvolvimento	A anexar

Atividades Comumente Realizadas na Organização:

Nome	Situação

Modelos de Documentos Previamente Definidos:

Nome	Situação
Especificação de Requisitos	A anexar

Procedimentos Previamente Definidos:

Nome	Situação

Figura VIII.4 – Tela de Identificação da Cultura Organizacional na Área de Software: Seção de Processos

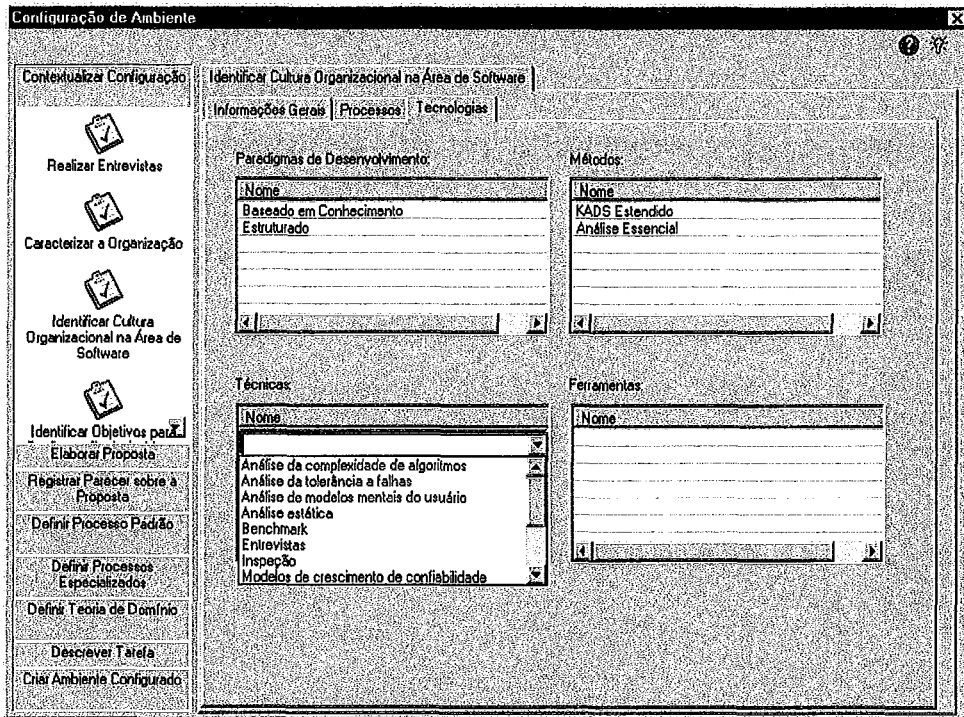


Figura VIII.5 – Tela de Identificação da Cultura Organizacional na Área de Software: Seção de Tecnologias

A figura VIII.6 exibe a tela *Identificar Objetivos para a Configuração do Ambiente*, na qual o configurador registra os objetivos da organização na área de software, identificando quais destes objetivos devem ser apoiados pelo ambiente a ser configurado. Exemplos de objetivos já foram mencionados na descrição do processo de configuração (anexo V).

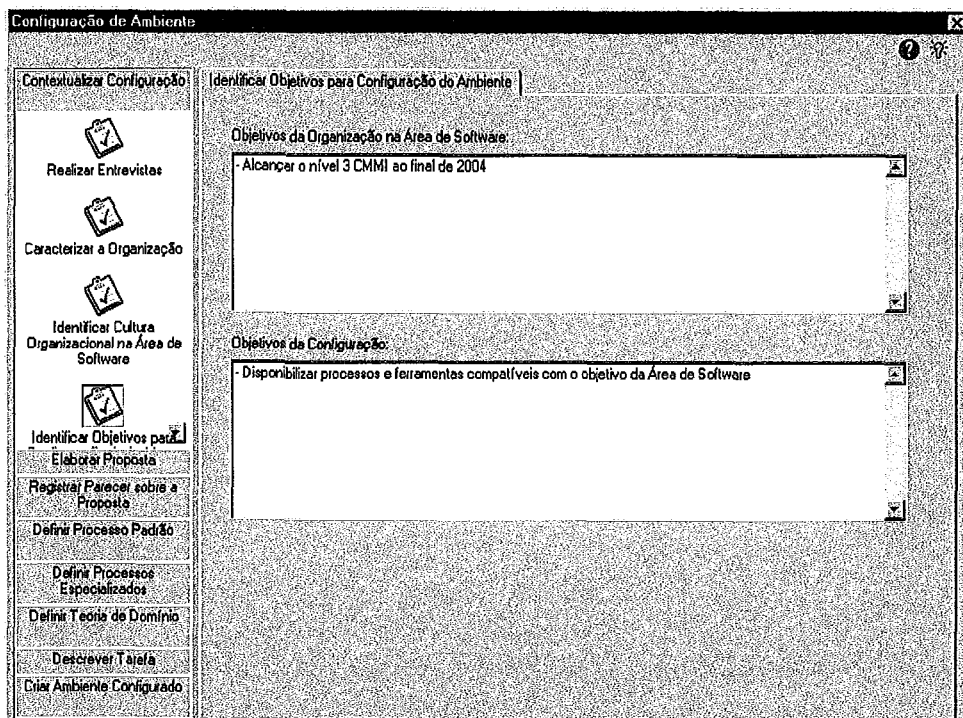


Figura VIII.6 – Tela de Identificação dos Objetivos para Configuração do Ambiente

A figura VIII.7 exibe a tela *Identificar Teorias de Domínio*, que é a primeira sub-atividade da atividade *Elaborar Proposta*. Nessa tela, o configurador de ambientes lista as Teorias de Domínio a serem disponibilizadas no Ambiente Configurado, informando se elas já se encontram definidas ou precisam ser definidas (campo *Estado Inicial*) e, quando já definidas, se há a necessidade de evolução das mesmas (campo *Observação*). Para cada Teoria do Domínio a ser definida ou evoluída, o configurador de ambientes pode explicitar o propósito, os cenários de motivação e as questões gerais de competência da Teoria do Domínio, estabelecendo o seu escopo e uso.

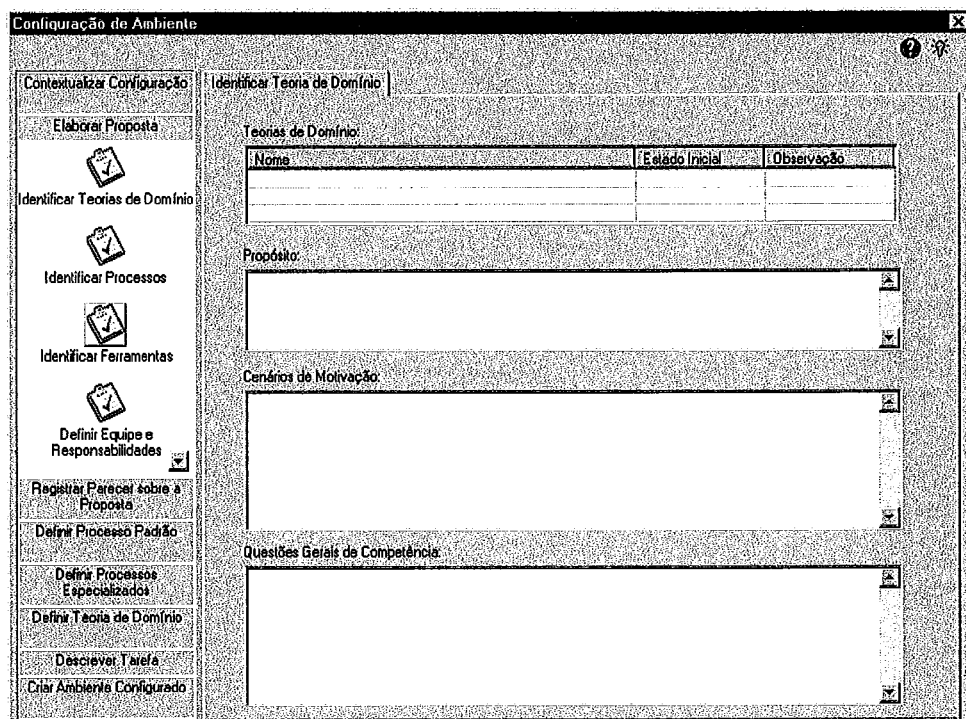


Figura VIII.7 – Tela de Identificação da Teoria do Domínio

A figura VIII.8 exibe a tela *Identificar Processos*, na qual o configurador de ambientes identifica os processos a serem disponibilizados no Ambiente Configurado, informando se eles já se encontram definidos ou precisam ser definidos (campo *Estado Inicial*) e, quando já definidos, se precisam ser revistos (campo *Observação*). Para auxiliar, a ferramenta fornece uma lista com os possíveis processos, combinando tipos de processos de ciclo de vida e paradigmas de desenvolvimento. O campo *Observação* também pode ser utilizado para indicar se os processos devem considerar atividades próprias de algum tipo de software. Além disso, para cada processo a ser definido ou revisto, o configurador de ambientes pode destacar aspectos que considere importantes para a definição/revisão do mesmo (campo *Descrição*).

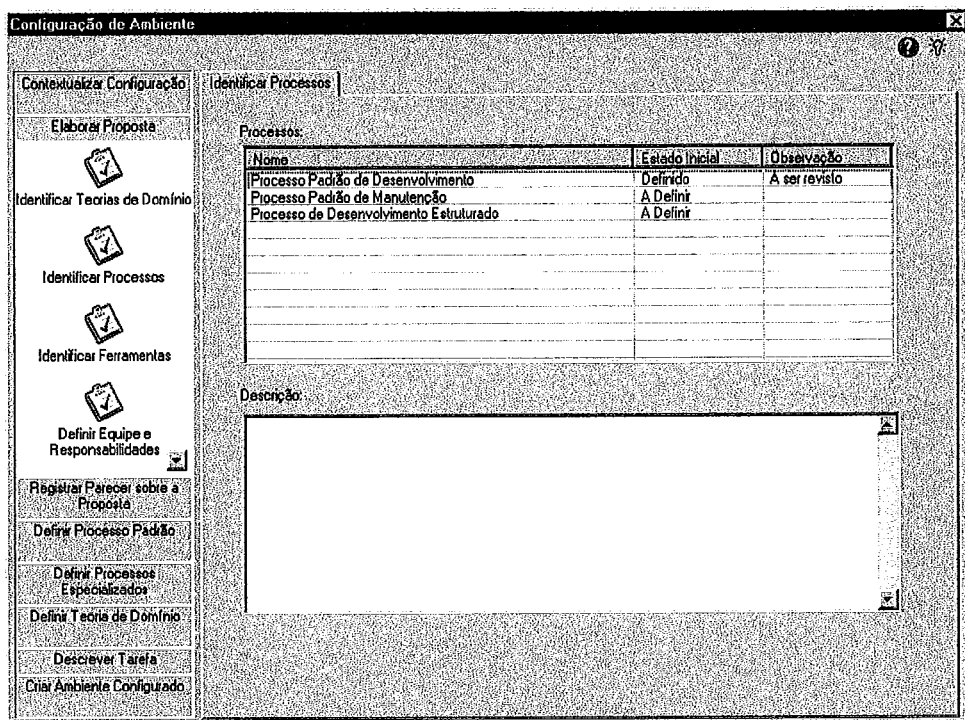


Figura VIII.8 – Tela de Identificação de Processos

Na tela *Identificar Ferramentas* (figura VIII.9), a ferramenta *Config* já exibe as ferramentas disponíveis na organização que foram informadas na atividade de *Identificar Cultura Organizacional na Área de Software*. O configurador de ambientes seleciona, então, as ferramentas disponíveis na Estação TABA a serem disponibilizadas no Ambiente Configurado e identifica, de acordo com os objetivos estabelecidos para a configuração, a necessidade de disponibilização de outras ferramentas, podendo, opcionalmente, propor a compra, desenvolvimento, ou obtenção de uma ferramenta *freeware* (campo *Conduta*).

As informações registradas nas telas da figura VIII.7, VIII.8 e VIII.9 são fundamentais para a definição da equipe, do cronograma e do custo da configuração do ambiente, que são os aspectos tratados nas telas a seguir.

A figura VIII.10, por exemplo, exibe a tela *Definir Equipe e Responsabilidades*, na qual o configurador de ambientes seleciona o coordenador do projeto de configuração do Meta-ambiente para a organização e estabelece as macro-atividades a serem executadas. Para cada macro-atividade, o configurador de ambientes pode definir uma equipe e um responsável pela execução. Para demonstrar a experiência das equipes e auxiliar na composição e justificativa do custo do projeto, a formação dos profissionais selecionados é exibida.

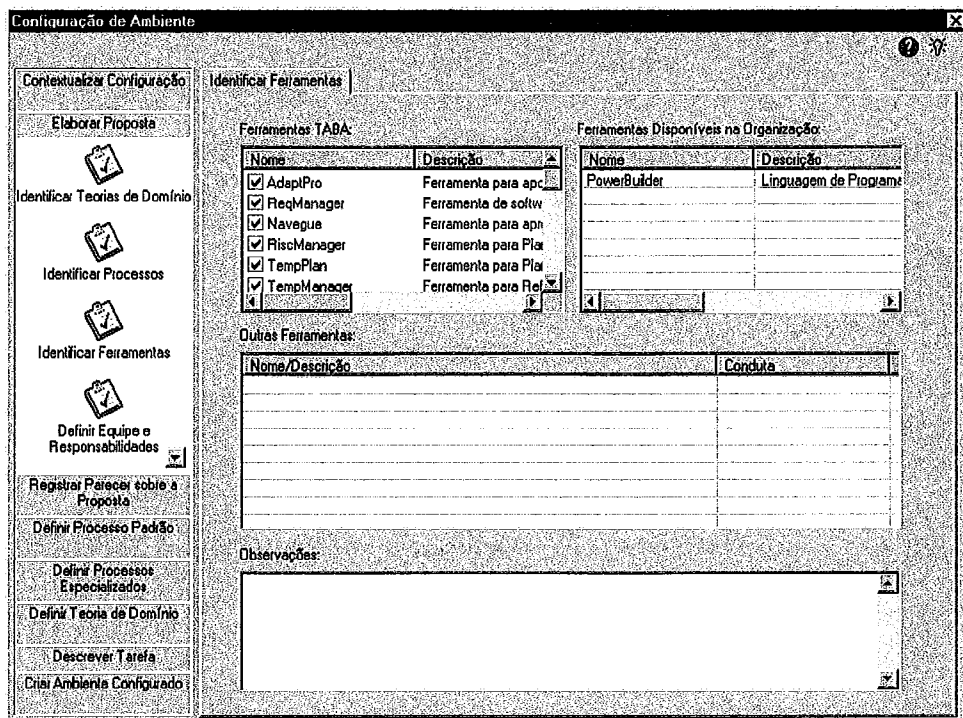


Figura VIII.9 – Tela de Identificação de Ferramentas

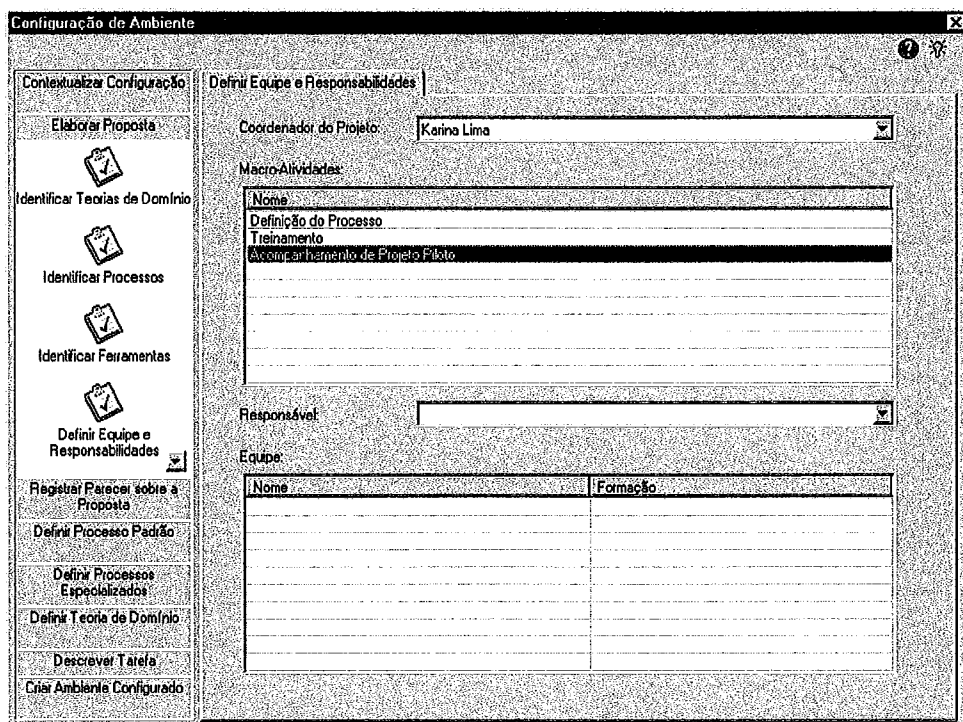


Figura VIII.10 – Tela de Definição de Equipes e Responsabilidades

Na tela *Definir Cronograma* (figura VIII.11), o configurador de ambientes define datas de início e fim para a execução de cada macro-atividade estabelecida na tela anterior. Na tela seguinte, *Definir Custos* (figura VIII.12), é a vez de definir o custo associado à cada macro-atividade, sendo o custo do projeto totalizado pela ferramenta.

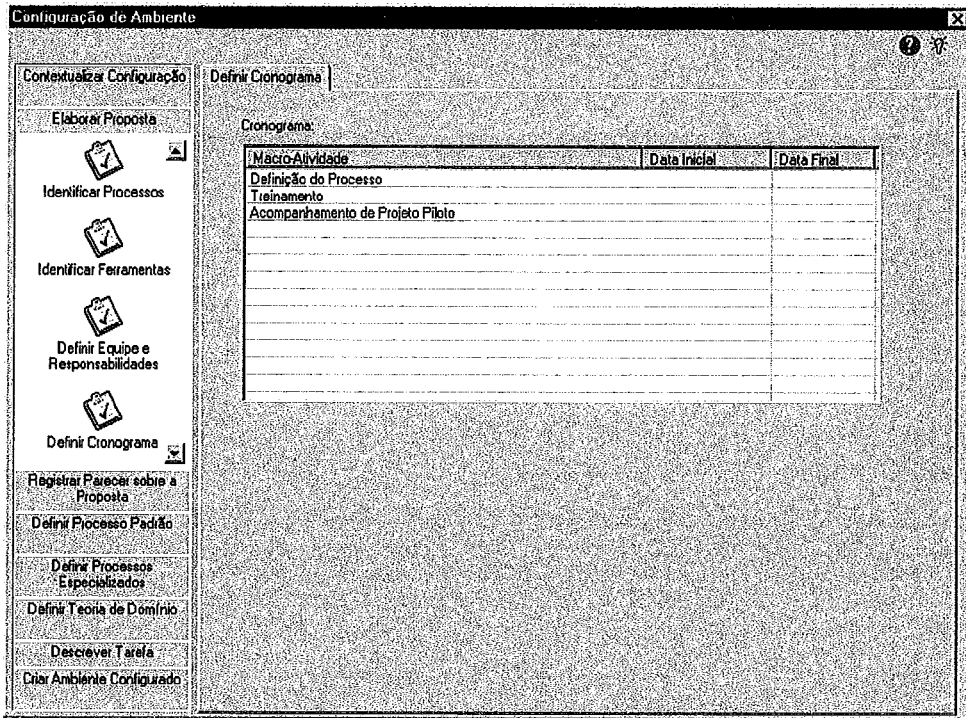


Figura VIII.11 – Tela de Definição de Cronograma

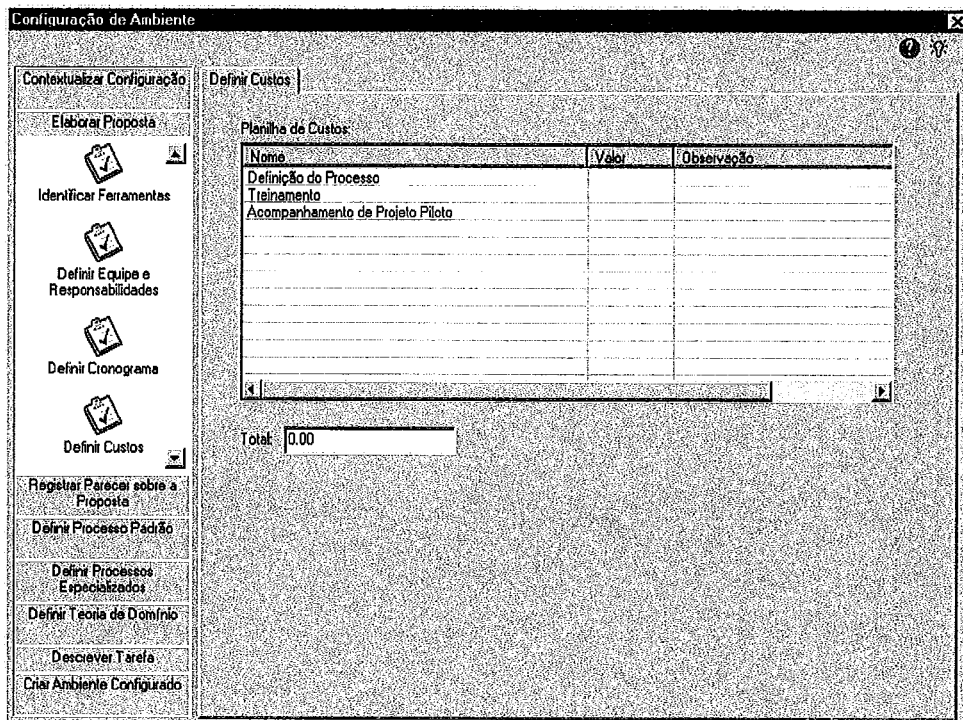


Figura VIII.12 – Tela de Definição de Custos

A figura VIII.13 exibe a tela *Enviar Proposta*, na qual o configurador de ambientes visualiza a proposta elaborada. Se verificar a necessidade de alteração de algum item, ele retorna a tela correspondente e efetua a alteração, retornando a tela *Enviar Proposta* para informar quem são os representantes da organização que devem receber a proposta, seus endereços de *e-mail* e o assunto a ser utilizado para a

postagem da mensagem. A ferramenta Config envia um e-mail com a proposta anexada para os representantes da organização, encerrando, assim, a atividade de *Elaborar Proposta*.

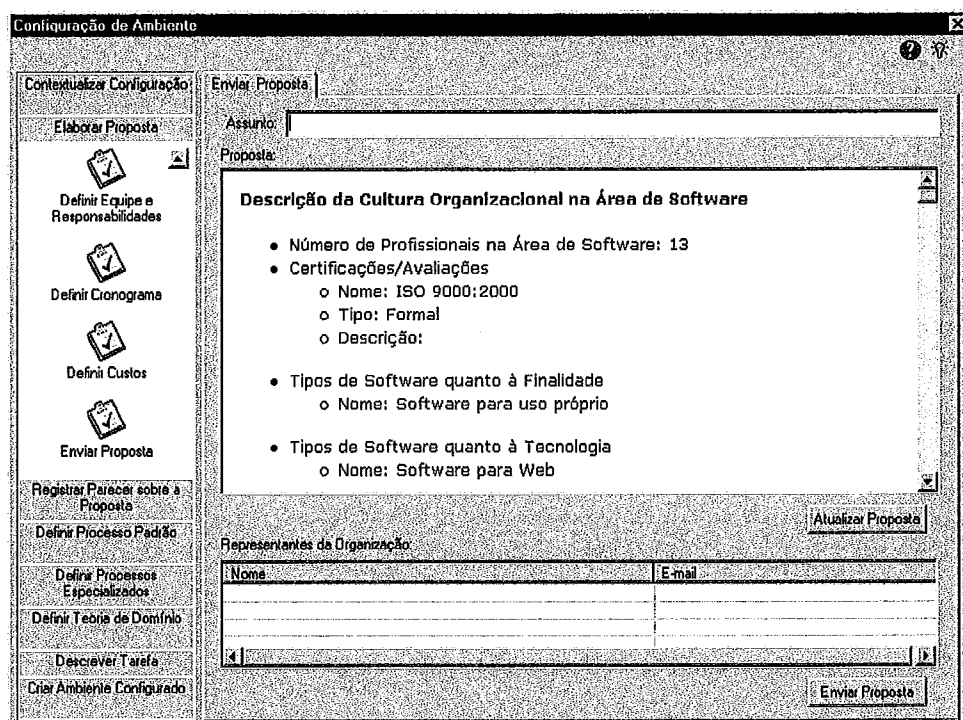


Figura VIII.13 – Tela de Envio de Proposta

A figura VIII.14 refere-se a tela *Registrar Parecer sobre Proposta*. Nesta tela, o configurador de ambientes informa-se sobre a data de envio da proposta e registra a data em que recebeu o parecer dos representantes da organização, qual foi este parecer e, quando pertinente, as modificações solicitadas e a data para envio da nova proposta. A depender do parecer, pode ser necessário retornar à atividade anterior.

Na tela exibida na figura VIII.15, *Caracterizar Processo Padrão*, o configurador de ambientes inicia a definição do processo padrão. Para isto, ele inclui um processo padrão e a ferramenta automaticamente inclui os processos de ciclo de vida já mencionados na proposta ou, caso não tenha sido elaborada uma proposta, inclui todos os processos de ciclo de vida contemplados na ferramenta. O configurador de ambientes pode, então, fornecer uma descrição para o processo padrão e indicar algum processo padrão, anteriormente definido, a ser utilizado com base para a definição de cada processo de ciclo de vida. Caso nenhum processo anteriormente definido seja utilizado com base para a definição dos novos processos de ciclo de vida, o configurador de ambientes especifica um padrão de referência, atualmente o ISO/IEC 12207 (1998), a ser utilizado para a definição.

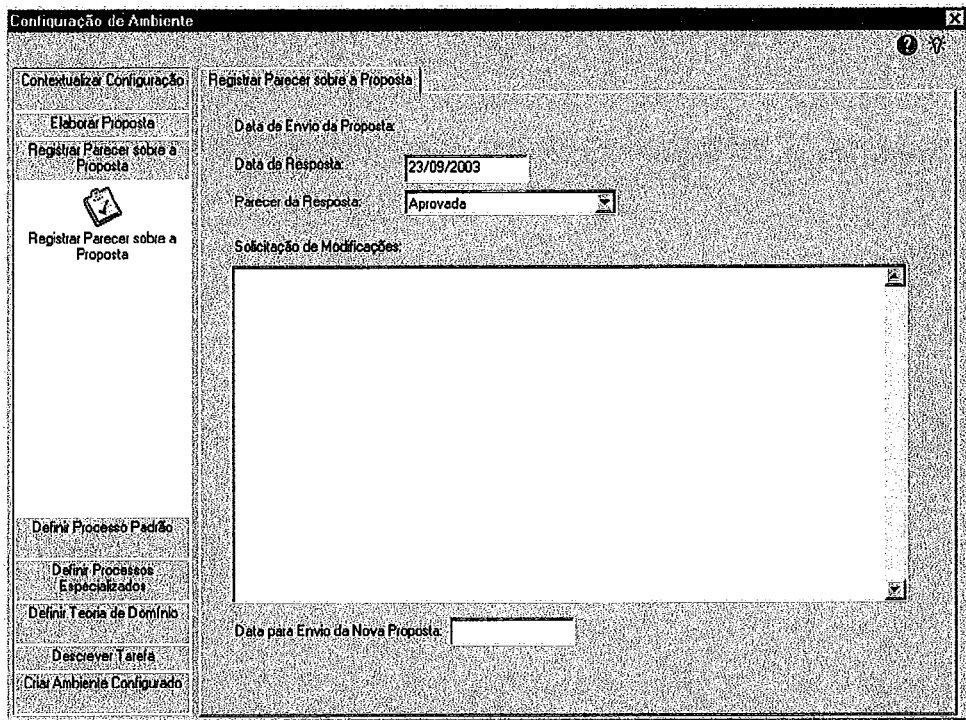


Figura VIII.14 – Tela de Registro do Parecer sobre a Proposta

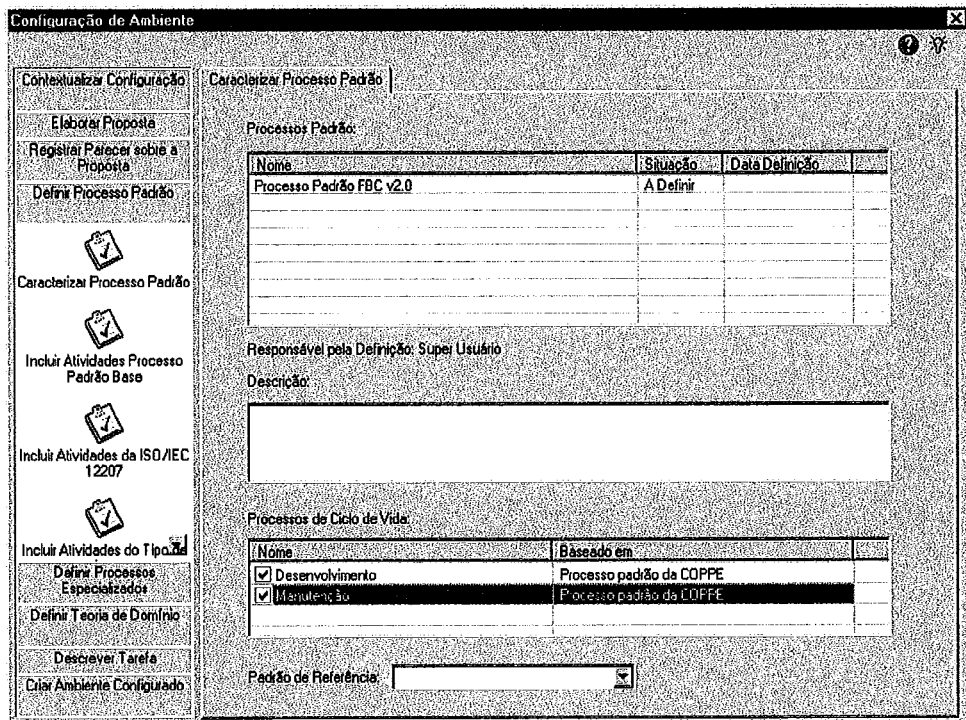


Figura VIII.15 – Tela de Caracterização do Processo Padrão

Na tela *Incluir Atividades do Processo Padrão Base* (figura VIII.16), o configurador de ambientes visualiza que, para cada processo de ciclo de vida, as atividades do processo padrão selecionado como base já foram incluídas, podendo utilizar a barra de ferramentas associada às atividades do processo para: alterar a descrição de uma atividade (📄), excluir uma atividade (🗑️), alterar a ordenação das

atividades (📄 e 📄), visualizar a descrição de uma atividade (📄), transformar uma atividade em sub-atividade da atividade seguinte (📄) ou transformar uma sub-atividade de uma atividade qualquer em uma atividade de mesmo nível e que a antecede (📄). Como não faz sentido a inclusão de novas atividades referentes a um processo base, o respectivo ícone da barra de ferramentas está desabilitado (📄). Já na barra de ferramentas do lado esquerdo, que é associada ao processo padrão utilizado como base, a única operação válida é visualizar a descrição de uma atividade. Cabe destacar que esta atividade só é executada se foi especificado algum processo padrão a ser utilizado como base para definição do processo padrão atual.

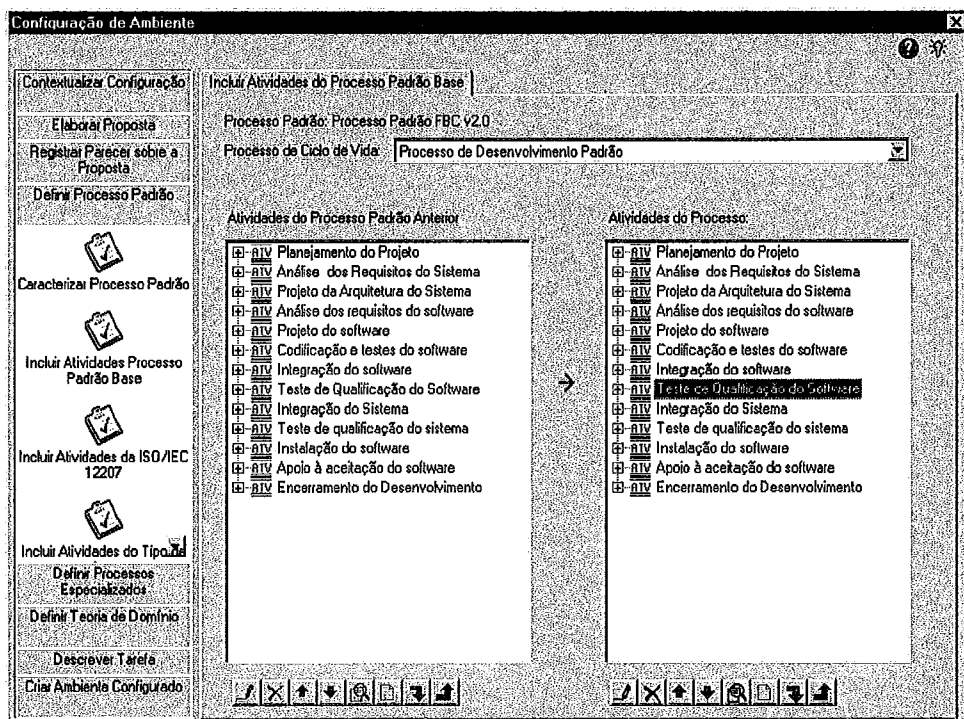


Figura VIII.16 – Tela de Inclusão de Atividades do Processo Padrão Base

As telas *Incluir Atividades da ISO/IEC 12207*, *Incluir Atividades do Tipo de Software*, *Incluir Atividades Próprias da Organização*, *Incluir Atividades Orientadas a Domínio* não são mostradas por serem semelhantes a tela da figura VIII.16, sendo que as atividades não são automaticamente incluídas no novo processo, devendo o configurador de ambientes, quando julgar necessário, selecionar as atividades que deseja incluir. Além disso, estas telas, a menos da que se refere às atividades ISO/IEC12207, permitem a inclusão de novas atividades.

Ainda com relação à definição de atividades, deve ser destacado que *Caracterizar projeto*, *Planejar processo* e *Instanciar ADSOrg* são atividades obrigatórias em qualquer processo padrão. Na figura VIII.16, tais atividades não aparecem porque foram incluídas como sub-atividades do Planejamento do Projeto.

Na tela *Estabelecer Atividades Obrigatórias* (figura VIII.17), o configurador de ambientes pode especificar quais atividades, além das já mencionadas, são obrigatórias e devem estar presentes em todos os processos definidos a partir do processo padrão e quais atividades podem ser retiradas de um processo específico a depender das características do projeto.

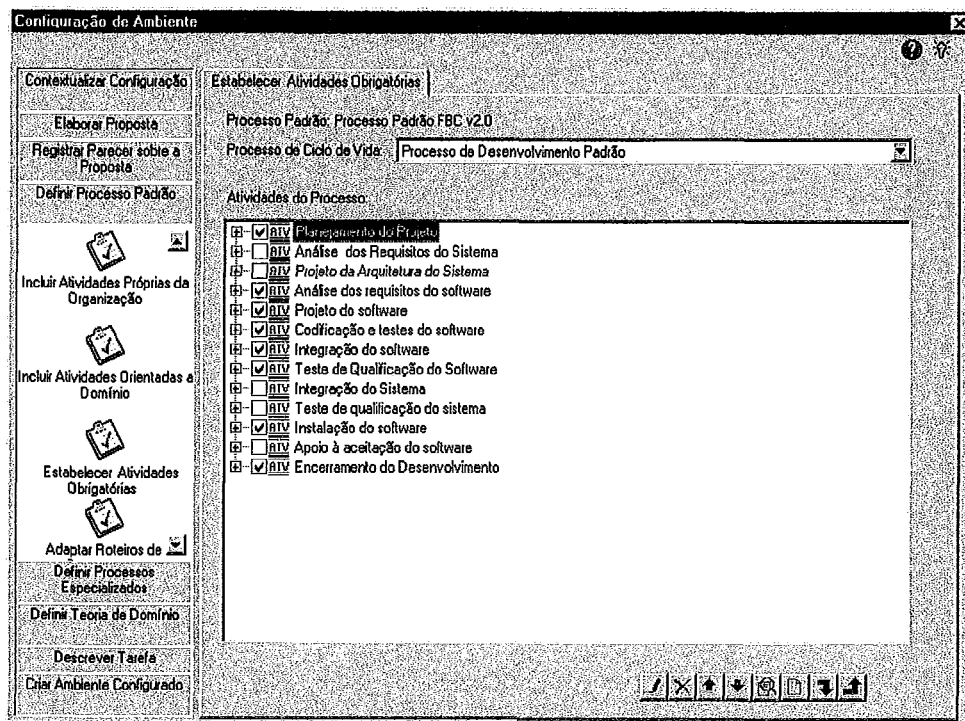


Figura VIII.17 – Tela para Determinação das Atividades Obrigatórias

A tela *Adaptar Roteiros* (figura VIII.18) permite que o configurador de ambientes adapte os roteiros de documentos disponíveis no Meta-ambiente para a processo sendo definido para a organização. Para isto, o configurador de ambientes seleciona uma atividade e um artefato da atividade, tendo acesso aos roteiros associados. Os roteiros obtidos durante a identificação da cultura organizacional (figura VIII.4) também podem ser visualizados e, se desejado, associados a um artefato do processo.

Na tela *Especializar Atividades do Processo Padrão* (figura VIII.19), o Config exibe os processos especializados a serem definidos, obtidos da proposta de configuração ou, caso uma proposta não tenha sido elaborada, dos paradigmas de desenvolvimento mencionados na atividade *Identificar Cultura Organizacional na Área de Software*. Quando o configurador de ambientes seleciona um processo especializado para definir, a ferramenta automaticamente especializa as atividades do processo padrão de acordo com o paradigma estabelecido. Novamente, a barra de ferramenta associada ao processo padrão (à esquerda) possibilita a visualização das atividades desse processo e a barra de ferramenta associada ao processo especializado (à direita) permite: a alteração e a visualização da descrição das

atividades; a alteração da ordenação das atividades; a exclusão de atividades; a transformação de atividades em sub-atividades e vice versa; além da inclusão de novas atividades. Caso seja necessário refazer a especialização, o configurador de ambientes pode solicitar a redefinição dos processos especializados (botão *Redefinir Processos Especializados*).

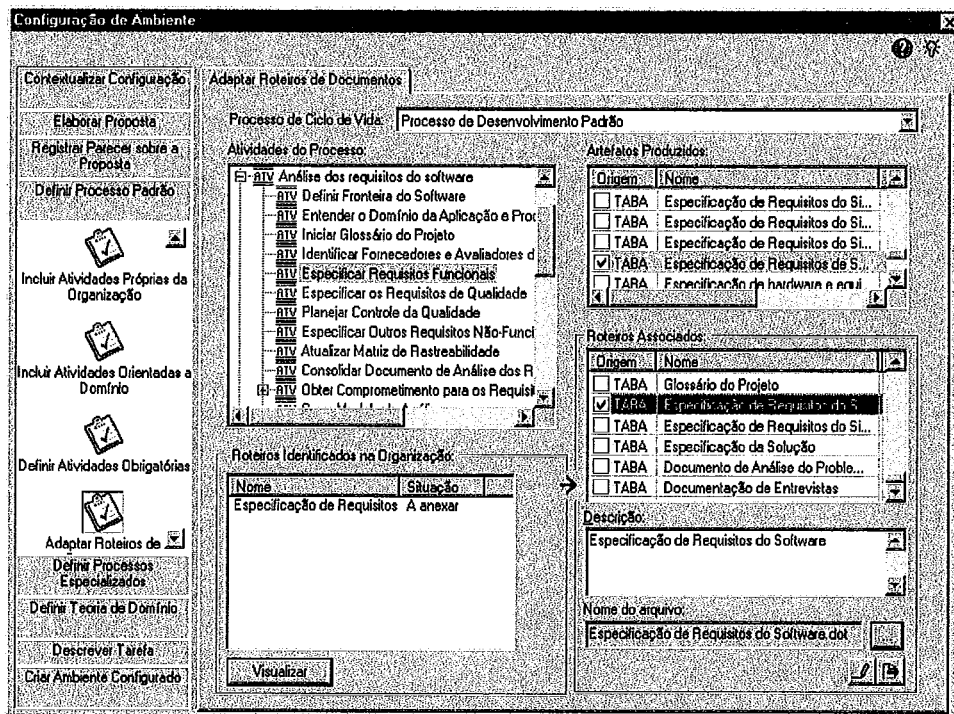


Figura VIII.18 – Tela de Adaptação de Roteiros de Documentos

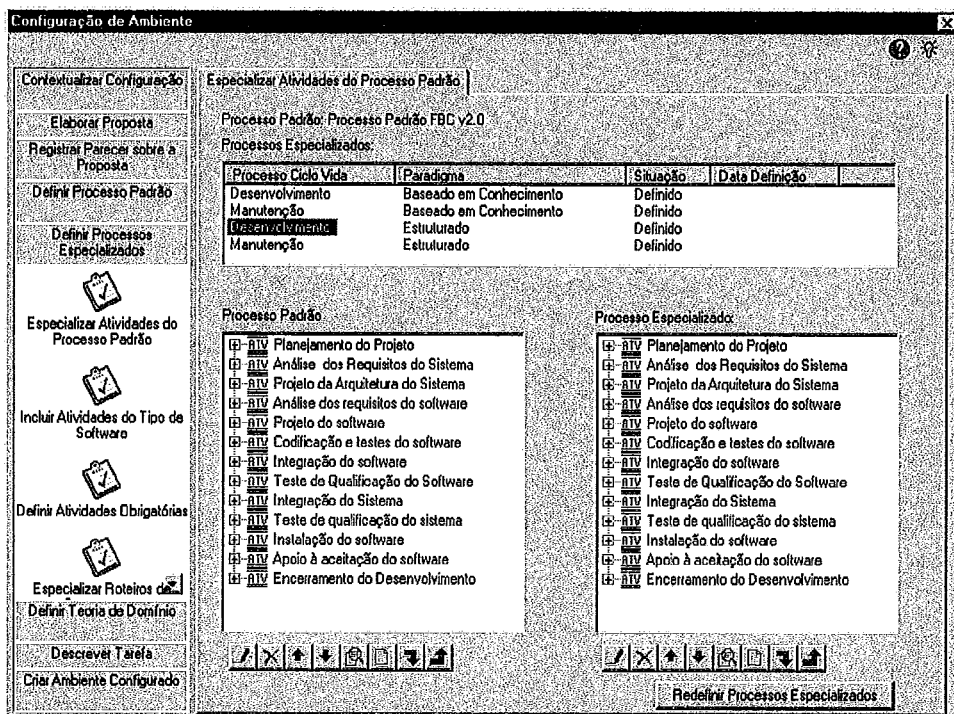


Figura VIII.19 – Tela de Especialização das Atividades do Processo Padrão

Como mencionado na descrição do processo de configuração, pode não ser necessário executar a atividade de *Incluir Atividades do Tipo de Software* nos processos especializados. Quando executada, a atividade é apoiada pelas funcionalidades disponíveis a partir da tela exibida na figura VIII.20. O configurador de ambientes seleciona um processo especializado para a inclusão de atividades que são próprias de um ou mais tipos de software, especificando, em seguida, o tipo de software a ser considerado. As atividades listadas para o tipo de software podem, então, ser incluídas no processo especializado. Caso o configurador de ambientes deseje obter processos especializados para o mesmo paradigma e diferentes tipos de software, ele inclui novos processos especializados, que herdam as atividades do processo especializado referente ao mesmo processo de ciclo de vida e paradigma, permitindo a inclusão das atividades próprias do tipo de software selecionado. O campo *Observação* é automaticamente atualizado com os tipos de software para o qual o processo foi especializado.

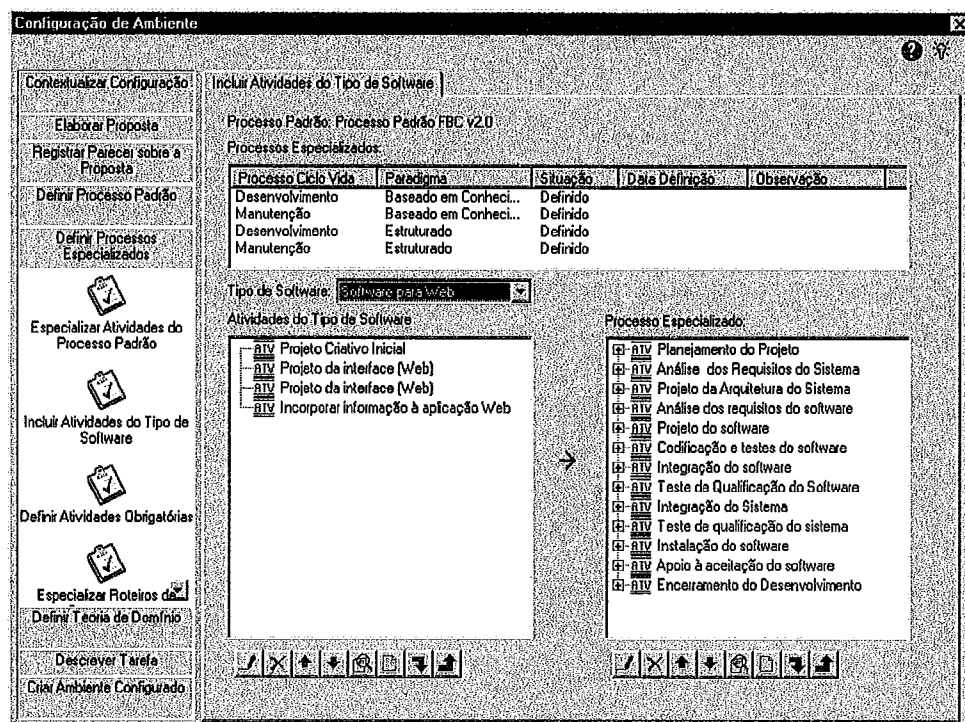


Figura VIII.20 – Tela de Inclusão de Atividades do Tipo de Software

A tela *Estabelecer Atividades Obrigatórias* para os processos especializados e a tela *Especializar Roteiros* não são exibidas por serem semelhantes às telas exibidas na figura VIII.17 e VIII.18, respectivamente.

A figura VIII.21 mostra a tela que apóia a atividade *Definir Teoria de Domínio*. Roteiros são fornecidos para auxiliar o configurador na produção dos artefatos *Definição do Escopo*, *Especificação de Requisitos*, *Captura e Avaliação da Teoria do*

Domínio. O artefato *Definição do Escopo* representa, na verdade, uma revisão do escopo previamente estabelecido na atividade *Identificar Teorias de Domínio*. A conceituação e formalização da Teoria do Domínio bem como a identificação das tarefas pertinentes e a implementação da Teoria do Domínio são apoiadas pela ferramenta EdiTeD [20,136].

A tela *Descrever Tarefa* não é apresentada por ser similar a tela *Definir Teoria de Domínio*, possibilitando o acesso a ferramenta EdiTar [137] e aos roteiros que auxiliam na descrição da tarefa e na sua avaliação.

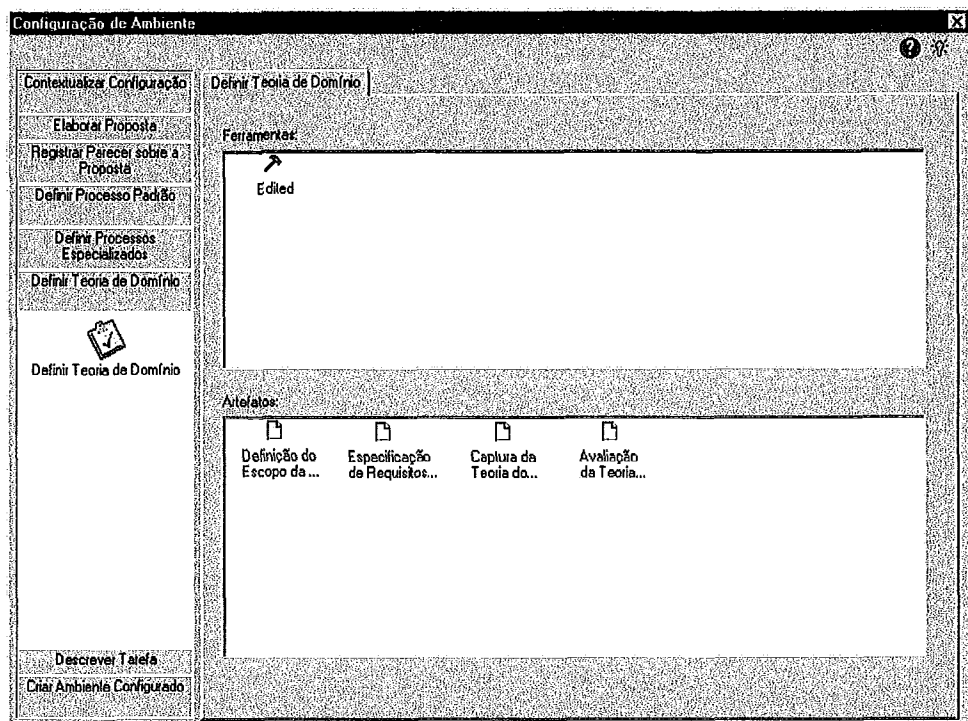


Figura VIII.21 – Tela para Definição da Teoria do Domínio

Na tela *Definir Ambiente*, exibida na figura VIII.22, o configurador de ambientes inicia a criação propriamente dita do Ambiente Configurado, fornecendo-lhe um nome e uma descrição e selecionando o processo padrão e as Teorias de Domínio a serem embutidas no ambiente. Na tela *Gerar Ambiente* (figura VIII.23), o configurador de ambientes seleciona os arquivos com a logomarca da organização e com o ícone a ser utilizado para acesso ao ambiente, informando, também, o diretório para o seu armazenamento. Por fim, na tela *Testar Ambiente* (figura VIII.24), o configurador de ambientes executa o Ambiente Configurado, selecionando-o de uma lista de ambientes gerados para a organização ou informando a sua localização.

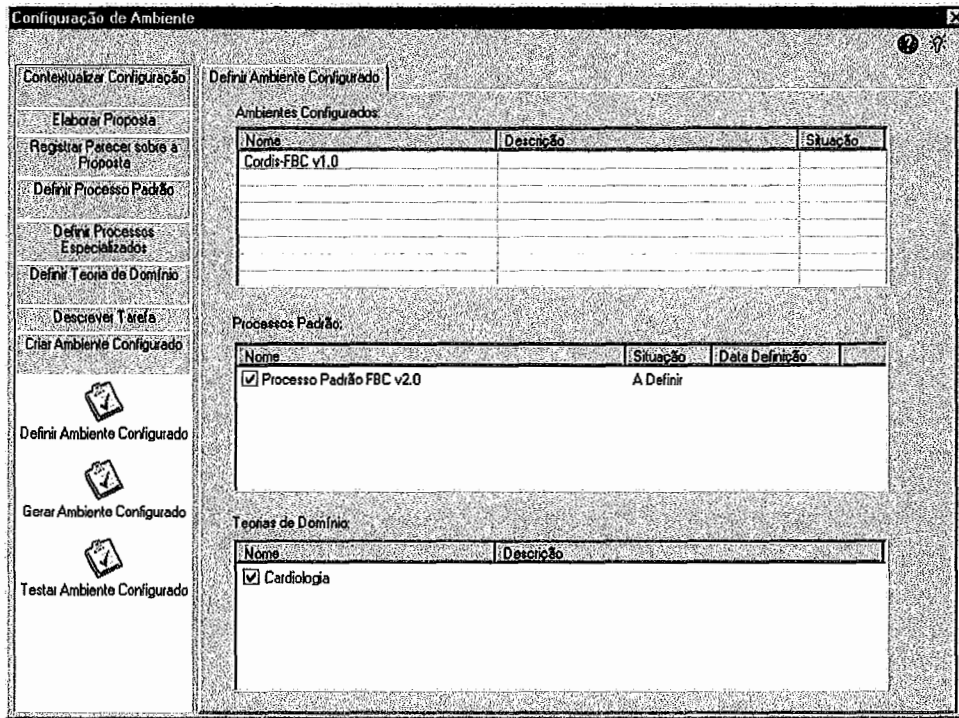


Figura VIII.22 – Tela para Definição do Ambiente Configurado

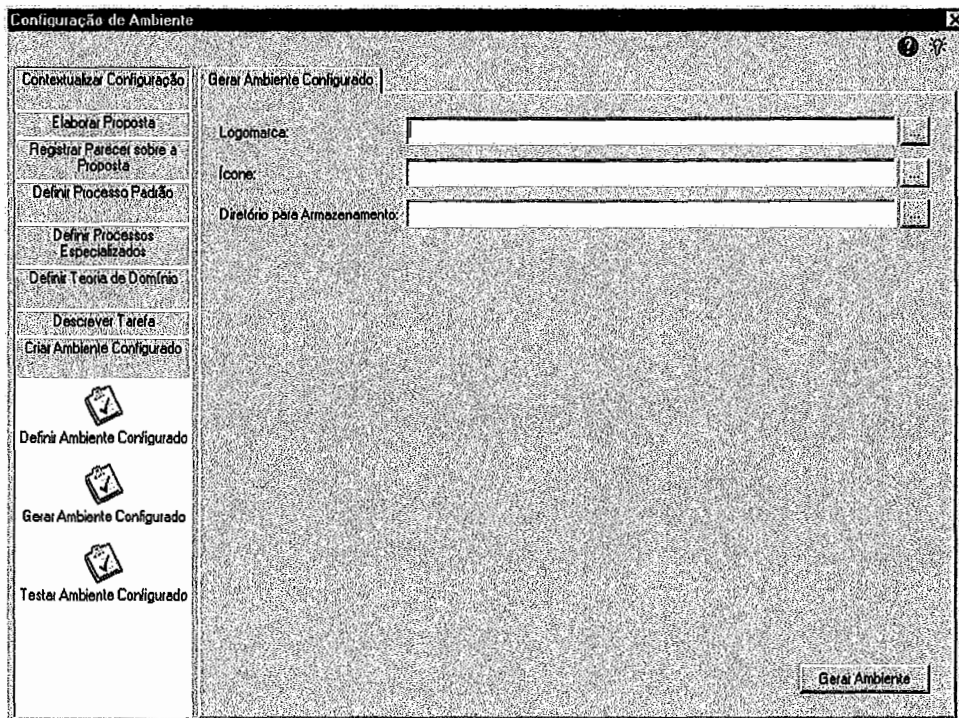


Figura VIII.23 – Tela para Geração do Ambiente Configurado

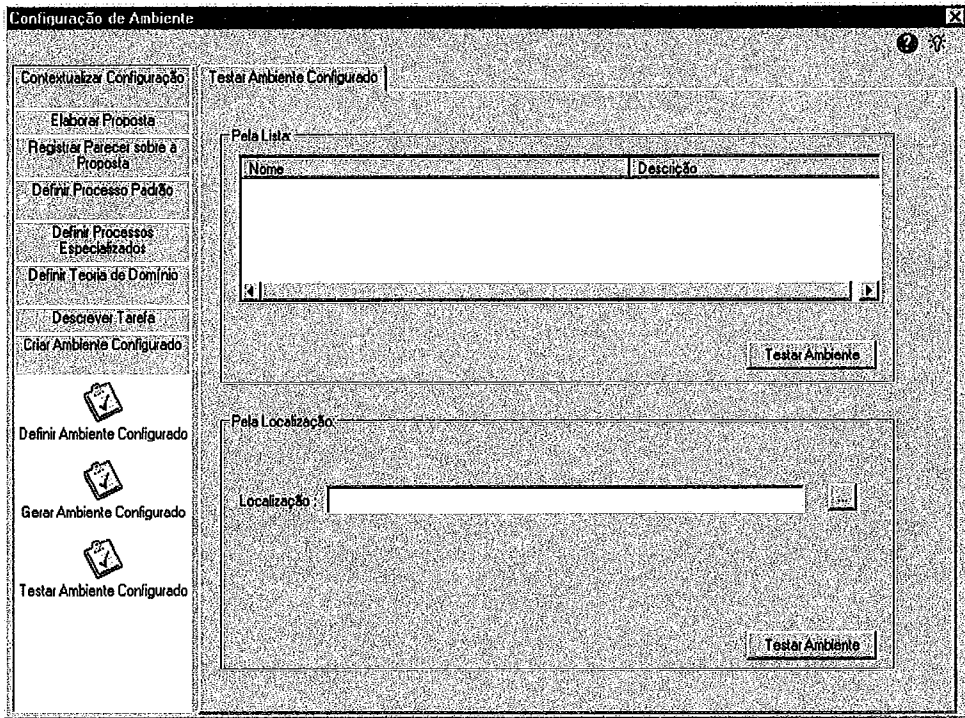


Figura VIII.24 – Tela para Teste do Ambiente Configurado

VIII.2 Descrição da Ferramenta Procknow

A figura VIII.25 mostra a tela principal da ferramenta Procknow, dividida em cinco seções e já inicialmente descrita no capítulo 7.

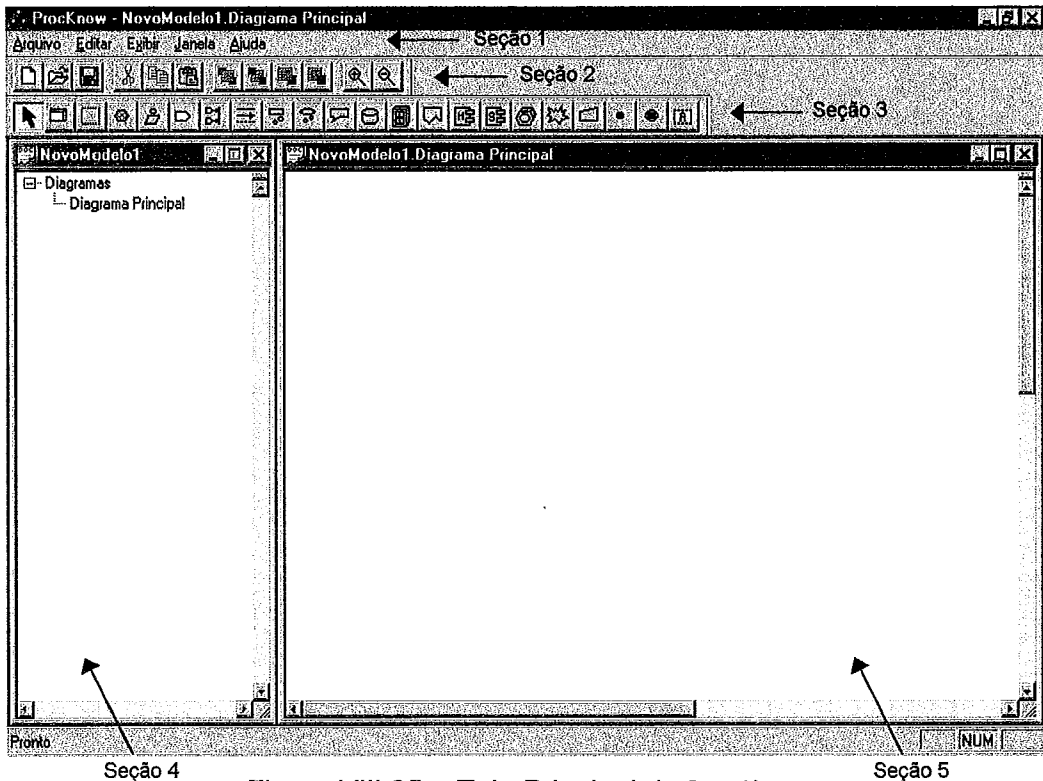


Figura VIII.25 – Tela Principal da Procknow

O primeiro *menu* (*menu* Arquivo) disponibiliza as seguintes operações: criar novo modelo, criar novo diagrama, abrir modelo já existente, fechar modelo selecionado, salvar modelo, salvar modelo com outro nome ou localização, configurar página para impressão, visualizar impressão, imprimir e encerrar execução da ferramenta. Através do *menu* Editar, o usuário pode: descrever um elemento gráfico, um diagrama ou o próprio modelo; mudar a posição do rótulo de um elemento gráfico; decompor um objeto gráfico decomponível (processo ou atividade composta); recortar, copiar, colar e excluir elementos gráficos do diagrama selecionado; selecionar todos os elementos gráficos de um diagrama; mover elementos gráficos para frente e para trás, além de avançá-los e recuá-los em relação a outros elementos gráficos. O *menu* Exibir permite que os elementos gráficos do diagrama sejam exibidos 30% maiores ou menores em relação ao seu tamanho atual e permite alternar entre o modo Normal e o modo de Visualização, controlando a possibilidade de edição do diagrama. O *menu* Janela permite a organização e seleção das janelas. Por fim, o *menu* Ajuda apenas informa sobre a finalidade e versão da ferramenta.

Na janela da seção 5, elementos gráficos podem ser inseridos em um diagrama livremente, desde que observadas as regras estabelecidas pela linguagem de modelagem de processos apresentada no capítulo 5 (seção 5.7.2). Ao selecionar um elemento gráfico e pressionar o botão direito do *mouse*, o usuário tem acesso rápido às operações de recortar, copiar, excluir e descrever. Caso o elemento gráfico já tenha sido descrito, a operação mudar a posição do rótulo também é disponibilizada. Além disso, a operação decompor é fornecida para processos ou atividades compostas.

Já na janela da seção 4, pressionar o botão esquerdo do *mouse* com um diagrama selecionado provoca a exibição do diagrama e pressionar o botão direito dá acesso rápido às operações de recortar, copiar, excluir e descrever.

As operações de descrever, mudar a posição do rótulo e decompor, por serem específicas da ferramenta, são exemplificadas nas figuras VIII.26 a VIII.35.

Ao descrever um objeto gráfico representado em um diagrama, o usuário pode selecionar um objeto lógico do mesmo tipo que já tenha sido definido ou pode definir um novo objeto lógico. Esta característica da ferramenta permite que um mesmo objeto lógico apareça em diferentes locais de um diagrama ou em diferentes diagramas, ou seja, que um objeto lógico tenha várias representações gráficas, o que é de fundamental importância para facilitar a diagramação dos elementos do modelo. Por exemplo, o conhecimento explícito *Regras do Convênio* pode ser um insumo para atividades que estão distantes no diagrama, requerendo a sua representação gráfica em dois locais diferentes. Para definir um novo objeto lógico, o usuário informa o nome do objeto, o rótulo a ser exibido no diagrama e uma descrição, além dos atributos

específicos do objeto, já apresentados no capítulo 5 (seção 5.7.2). Para atividades compostas ou processos, a descrição também inclui o campo *Decomposição*, que permite a associação de um diagrama já existente ao objeto sendo descrito, representando a sua decomposição em (sub-)atividades. As figuras VIII.26 a VIII.28 ilustram a descrição de um repositório, de um conhecimento explícito e de uma atividade.

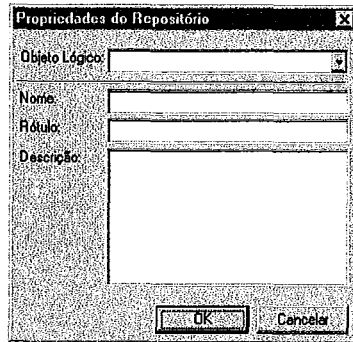


Figura VIII.26 – Janela para Descrição de Repositório

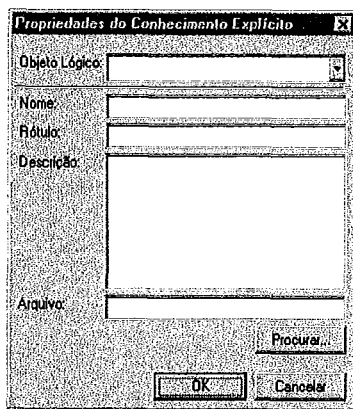


Figura VIII.27 – Janela para Descrição de Conhecimento Explícito

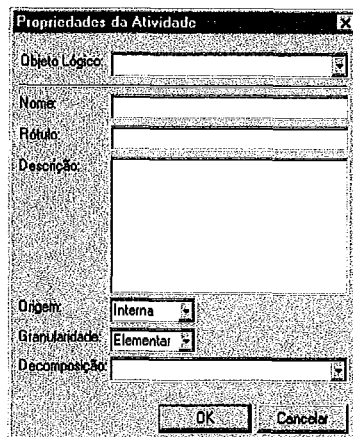


Figura VIII.28 – Janela para Descrição de Atividade

A operação descrever também é disponibilizada para as ligações gráficas que admitem atributos especiais, que são as que representam Fluxo de Controle e

Informações, Fluxo de Entrada/Saída para um Armazenador (que pode ser um objeto Repositório ou Arquivo) ou uma Associação Não Direcionada envolvendo um objeto Evento. As figuras VIII.29 a VIII.31 mostram as janelas para descrição dessas ligações.

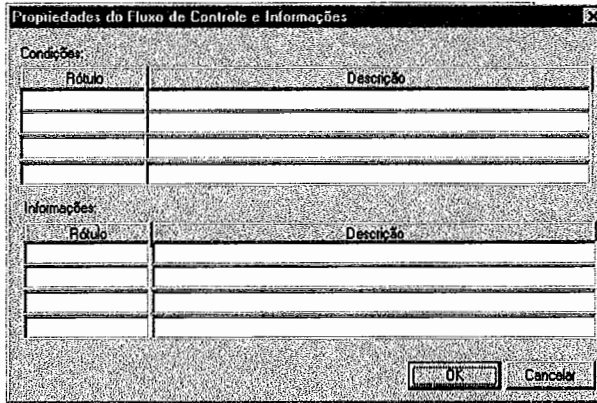


Figura VIII.29 – Janela para Descrição de Fluxo de Controle e Informações

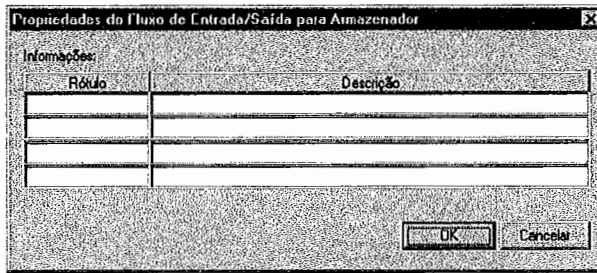


Figura VIII.30 – Janela para Descrição de Fluxo de Entrada/Saída para um Armazenador

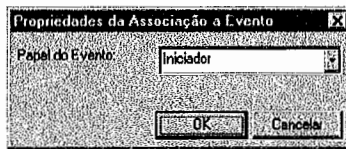


Figura VIII.31 – Janela para Descrição de Associação a Evento

Quando o usuário aciona a operação descrever para um diagrama, são solicitados um nome e uma descrição (figura VIII.32). Já a operação de descrever modelo, disponível no *menu Editar*, permite que seja especificada uma organização e uma versão para o modelo (figura VIII.33).

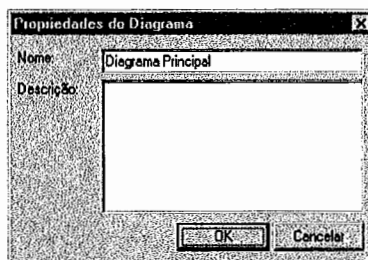


Figura VIII.32 – Janela para Descrição de Diagrama

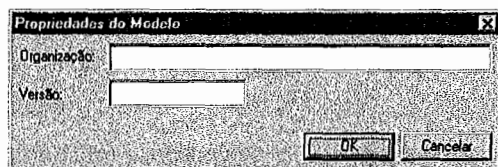


Figura VIII.33 – Janela para Descrição de Modelo

A operação mudar posição do rótulo, permitida para um objeto gráfico já descrito, aciona a janela da figura VIII.34.

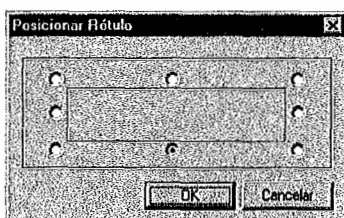


Figura VIII.34 – Janela para Posicionamento do Rótulo

Já a operação decompor é disponibilizada a partir da seleção de uma atividade composta ou de um processo que já tenham sido descritos (figura VIII.35). Quando acionada, a operação provoca a criação de um novo diagrama já associado ao objeto selecionado, que é aberto para edição em uma nova janela da seção 5 da ferramenta.

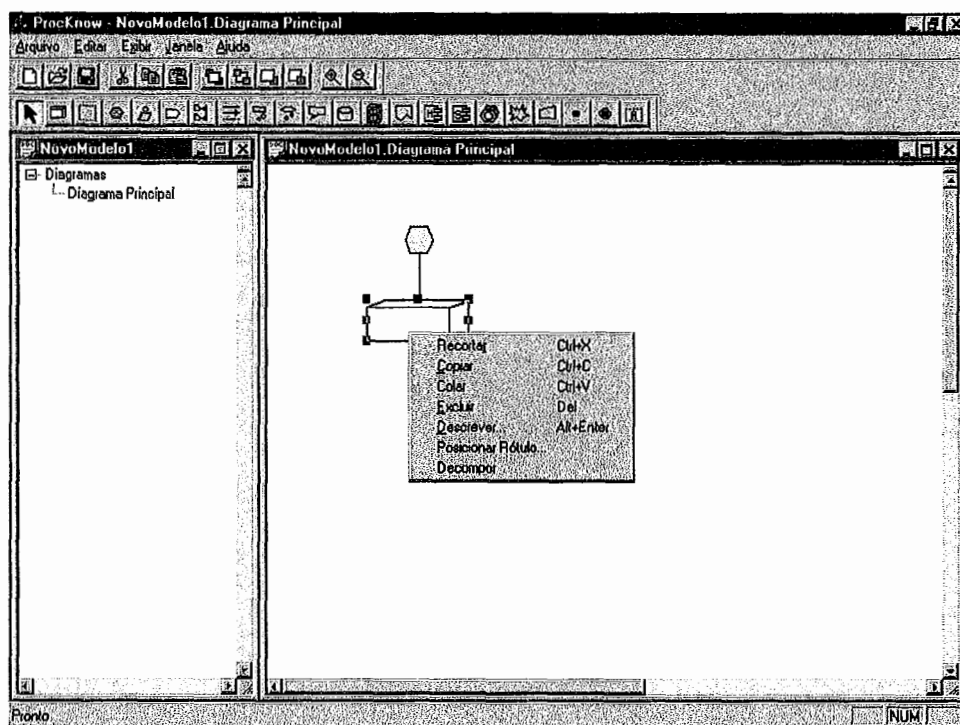


Figura VIII.35 – Menu de operações disponíveis para um processo

Além disso, quando o usuário insere um Ator ou uma Atividade dentro de uma Área de Ator, estes objetos são automaticamente vinculados à Área de Ator. O mesmo acontece quando um Processo é inserido em um Grupo de Processos. Ao mover-se

um objeto, move-se os objetos vinculados ao mesmo, o que não impede que atividades e processos sejam movidos para fora das áreas a que pertencem, desvinculando-se dos objetos de área correspondentes. Quando áreas são redimensionadas e não mais englobam as mesmas atividades ou processos, também é feita a desvinculação dos objetos. Antes de executar as operações de mover, redimensionar, excluir, recortar e colar, a ferramenta verifica se as regras da linguagem não vão ser violadas. Por exemplo, o último Ator de uma Área de Ator não pode ser excluído ou recortado, por exemplo. A figura VIII.36 ilustra a mensagem de aviso emitida pela ProKnow.



Figura VIII.36 – Mensagem de Violação das Regras da Linguagem

ANEXO IX

Modelos de Classes da Ferramenta ProKnow

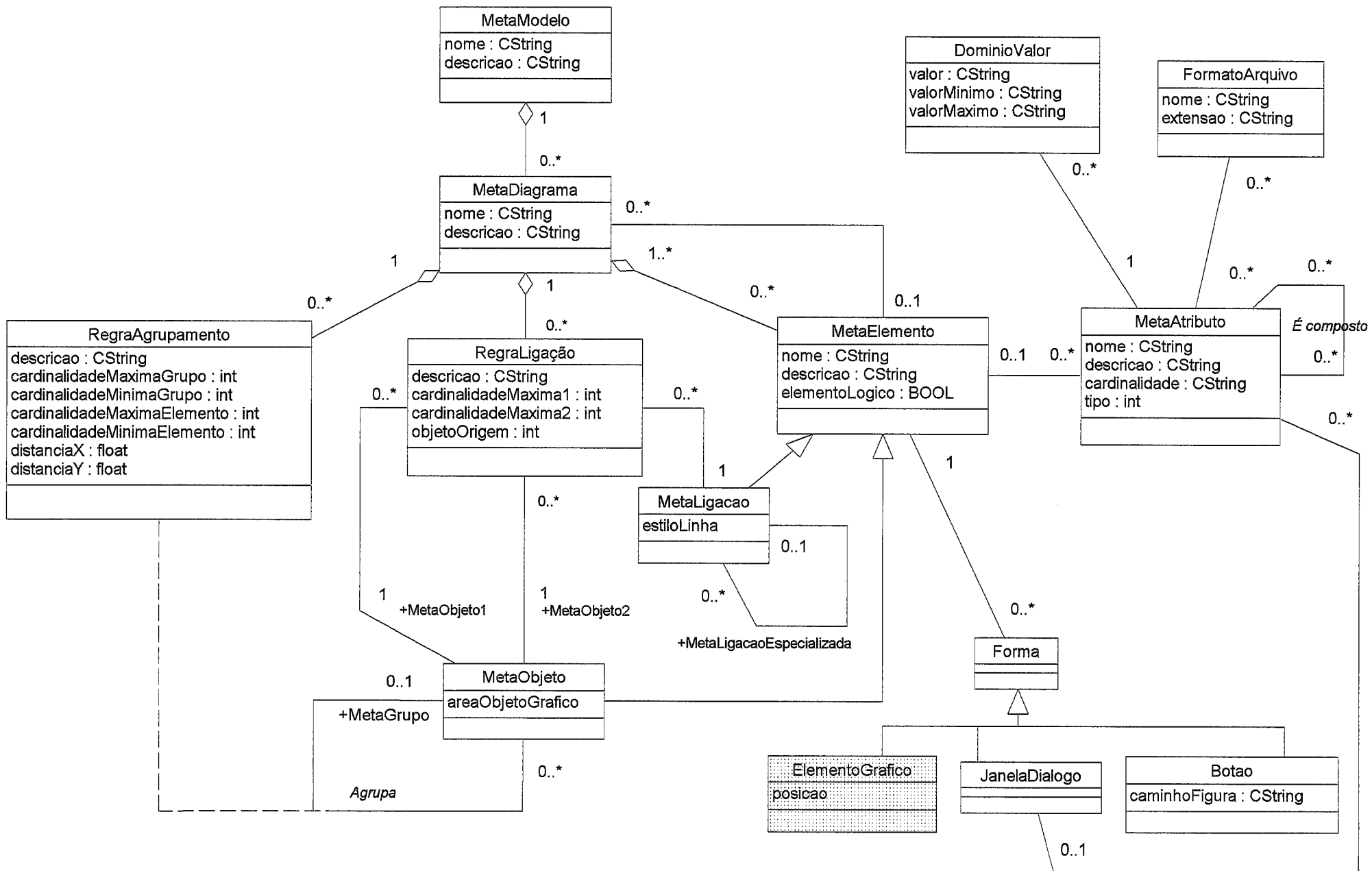


Figura IX.1 - Classes Genéricas da Ferramenta ProckKnow (Nível Meta-modelo)

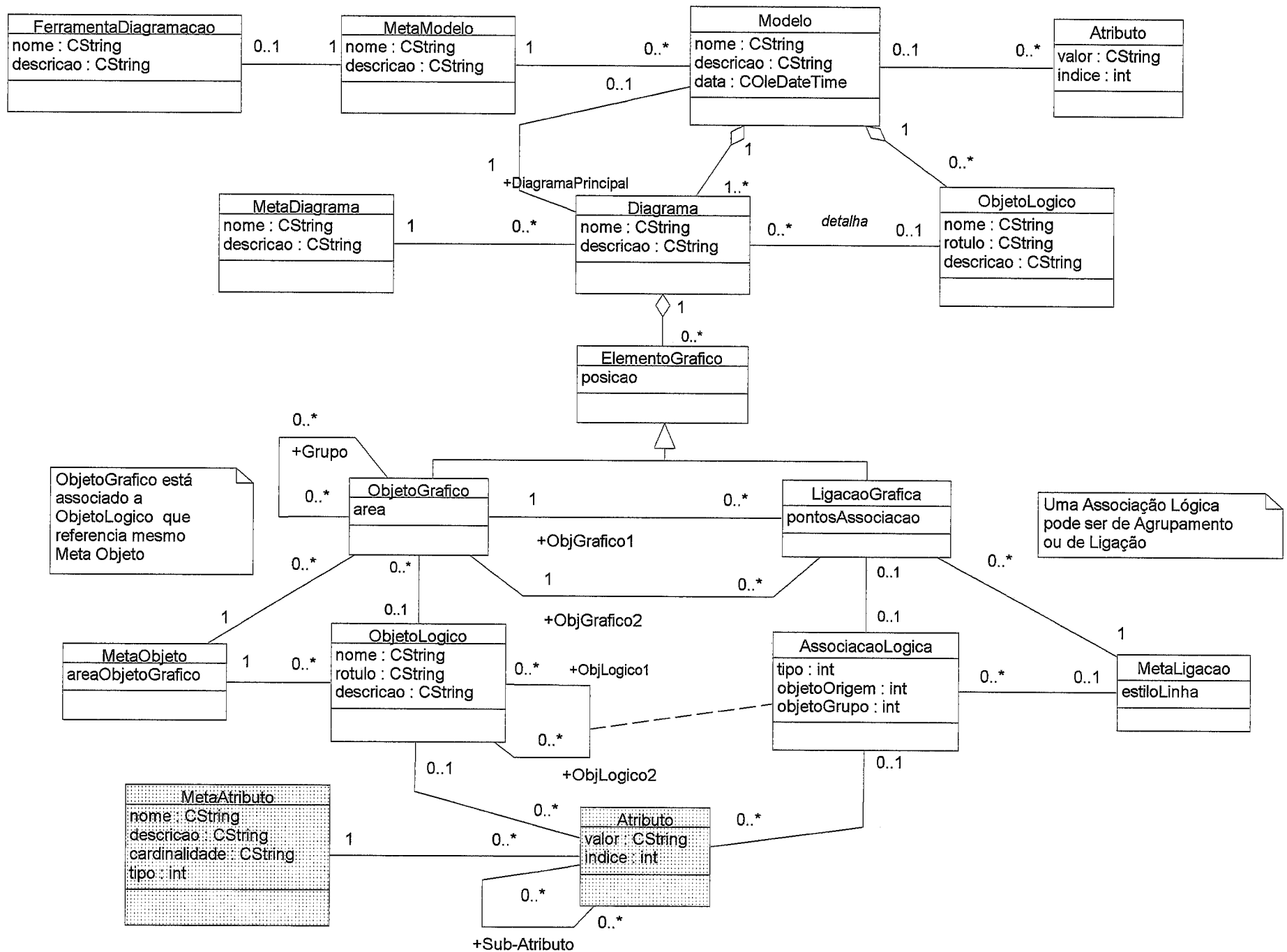


Figura IX.2 - Classes Genéricas da Ferramenta ProckNow (Nível Modelo)

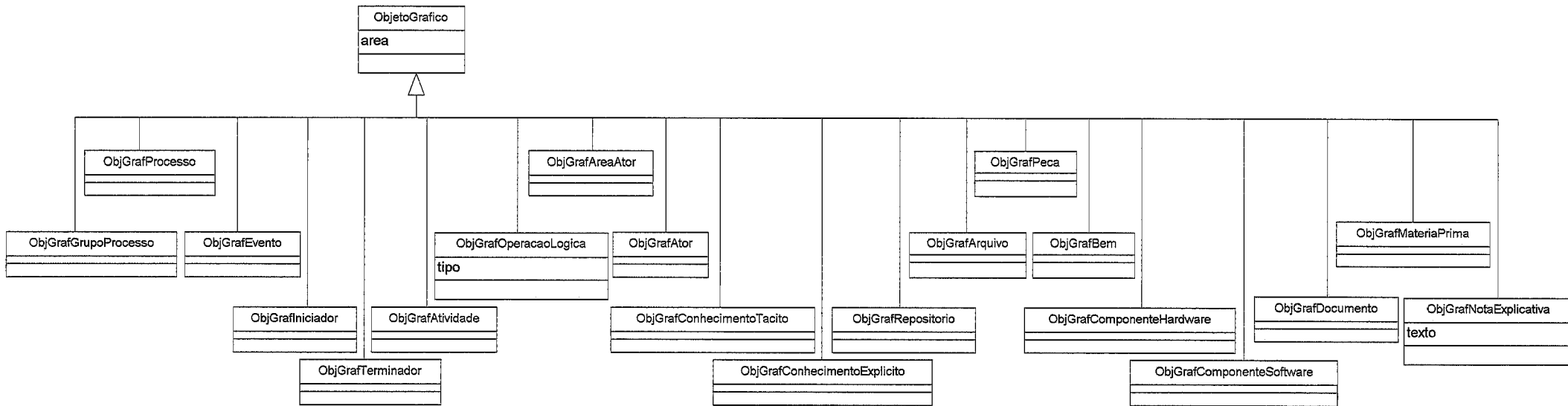
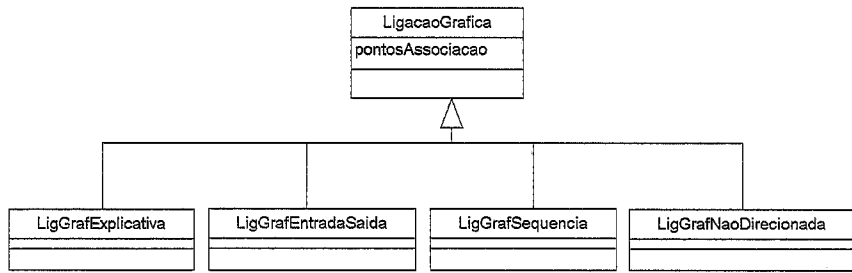


Figura IX.3 - Classes Especificas da Ferramenta Procknow