

# Definição de Processos em Ambientes de Desenvolvimento de Software Orientados a Organização

Karina Villela, Gleison Santos, Mariano Montoni, Patrícia Berger, Sávio Figueiredo, Sômulo Mafra, Ana Regina Rocha, Guilherme Horta Travassos

COPPE/UFRJ – Programa de Engenharia de Sistemas e Computação  
Caixa Postal 68511 – CEP: 21945-970

Rio de Janeiro – RJ

[kvillela@cos.ufrj.br](mailto:kvillela@cos.ufrj.br), [gleison@cos.ufrj.br](mailto:gleison@cos.ufrj.br), [darocha@cos.ufrj.br](mailto:darocha@cos.ufrj.br)

## Resumo

Ambientes de Desenvolvimento de Software têm evoluído ao longo do tempo para fornecer apoio mais amplo e efetivo aos desenvolvedores de software. Neste contexto, surgiram os Ambientes de Desenvolvimento de Software Centrados em Processo e os Ambientes de Desenvolvimento de Software Orientados a Organização. Este artigo apresenta a abordagem adotada pela Estação TABA, um meta-ambiente de desenvolvimento de software, para a definição de processos de software para organizações e projetos específicos dentro do contexto dos Ambientes de Desenvolvimento de Software Orientados a Organização. Esta abordagem está baseada na definição de processos em três níveis: processos padrão, especializado e instanciado.

**Palavras-chave:** Processo de Software, Qualidade de Software, Ambiente de Desenvolvimento de Software

## Abstract

Software Development Environments have evolved to support software developers more effectively. In this context, have appeared the Process-centered Software Development Environments and the Enterprise-Oriented Software Development Environments. This paper presents the approach adopted on the TABA Workstation, a software development meta-environment, for the software process definition for organizations and specific projects in the context of Enterprise-oriented Software Development Environments. This approach is based upon the three level process definition: standard, specialized, and instantiated processes.

**Keywords:** Software Process, Software Quality, Software Development Environment

## 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 [1] e, mais recentemente, a idéia de integrar Gerência do Conhecimento em ADS e os Ambientes de Desenvolvimento de Software Orientados a Organização (ADSOrg) [2].

A Estação TABA é um meta-ambiente de desenvolvimento de software capaz de gerar processos de desenvolvimento de software centrados em processo para organizações (os Ambientes Configurados) e/ou projetos específicos (os ADSOrg). 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. Este processo considera a definição de um Processo Padrão para a organização para a qual o ambiente será construído, que, por sua vez, serve 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 [1].

A primeira etapa da definição dos processos consiste, então, 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. A segunda envolve a criação de um processo especializado para os diferentes paradigmas e métodos de desenvolvimento de software, além de características próprias da organização. A terceira etapa envolve a adaptação dos processos especializados para a realidade de projetos específicos levando em consideração modelos de ciclo de vida, características do projeto e da equipe, disponibilidade de recursos e requisitos de qualidade do produto. Os processos de software Padrão e Especializados são utilizados para configuração de ambientes específicos para uma organização, os Ambientes Configurados, enquanto o processo de software instanciado é utilizado para instanciar um ambiente para um projeto específico, os ADSOrg.

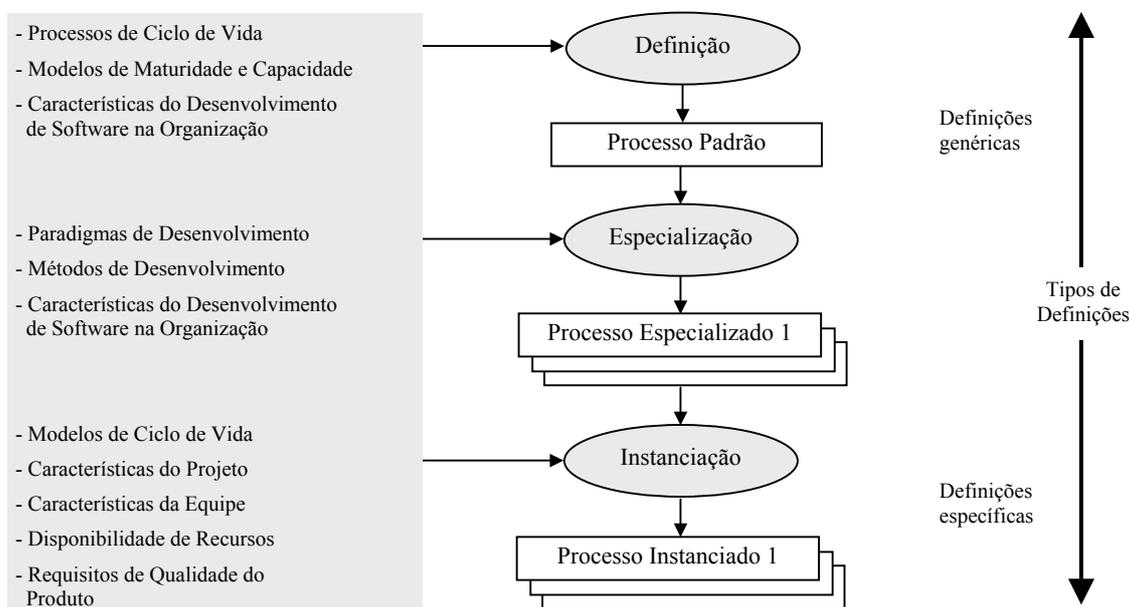
Este artigo apresenta a abordagem adotada pela Estação TABA para a definição de processos de software para organizações e projetos específicos dentro do contexto dos Ambientes de Desenvolvimento de Software Orientados a Organização. Na seção 2 serão discutidos modelos de processo de software e a abordagem de definição de processos em três níveis (processos padrão, especializado e instanciado) adotada. Na seção 3 são apresentadas as ferramentas construídas para dar suporte a esta abordagem. Por fim, a seção 4 apresenta as considerações finais e perspectivas futuras.

## 2. Modelos de Processo de Software

EMAN [3] 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. O nível *Definição de Processo*, foco desta seção, 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. A abordagem para definição de processos proposta neste artigo combina, de maneira sistemática e incremental, a maioria dos elementos mencionados por EMAN [3].

A Figura 1 representa, de forma esquemática, a abordagem adotada. Segundo a proposta do modelo, que estabelece etapas e produtos intermediários, a Norma ISO/IEC 12207 [4] é a base para a definição de qualquer Processo Padrão a partir da Estação TABA. 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 a objetos 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.



**Figura 1 - Abordagem para Definição de Processos**

A abordagem adotada, 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 armazenamento de conhecimento especializado sobre o desenvolvimento e a manutenção de software na organização e fornecer este conhecimento para as equipes de projeto quando necessário. Neste 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 [5] e o ISO/IEC 15504 [6] 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) [7] faz referência a um conjunto de processos padrão da organização, que contêm as definições de processo que guiam todas as atividades da organização, cobrindo todos os processos necessários na organização e em seus projetos.

## 2.1 O Processo Padrão

Um elemento chave das normas e modelos de maturidade é a definição de um processo padrão que descreve as atividades que devem ser realizadas no desenvolvimento de sistemas de software em todos os projetos de uma organização. O uso de modelos de processo genéricos como base para o planejamento do processo de software específico para um projeto permite aos gerentes definir planos em conformidade com os padrões de qualidade e procedimentos da organização [8, 9].

Na definição de um processo padrão para uma organização, é necessário considerar as características da organização [1]. Tais características envolvem: atividades realizadas nos projetos de software da organização não provenientes de um modelo proposto na literatura, que são consideradas atividades próprias da organização; o fato de a organização desenvolver

software para uso próprio, para comercialização ou sob encomenda; o fato de a organização ser especializada no desenvolvimento de um tipo de software específico; os problemas cruciais enfrentados pela organização em seus projetos de software, dentre outras.

Segundo EMAM et al. [10], 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.

OLIVEIRA [1] propõe a utilização de Processo Padrão 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 [1] destaca como premissas:

- A norma ISO/IEC 12207 estabelece os processos do ciclo de vida de um software para um projeto genérico e deve ser adaptada para organizações e projetos específicos. Para esta adaptação, é necessário identificar as características do ambiente do projeto;
- É preciso considerar a cultura organizacional, 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 [4, 10].

Considerando a importância de padronizar, e 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, por exemplo, é mais fácil desconsiderar a descrição de uma atividade não obrigatória do que elaborar uma descrição para a mesma.

Para a elaboração de um processo padrão para a organização é preciso dispor de conhecimento sobre as atividades de desenvolvimento e manutenção de software que são sempre realizadas na organização independente de um projeto específico. A definição do processo padrão considera as características de desenvolvimento de software na organização. Assumindo que os principais objetivos das organizações produtoras de software são definir, utilizar e estabelecer melhorias contínuas nos seus processos de software, torna-se fundamental para alcançar estes objetivos que tais organizações direcionem seus esforços não

só para a aplicação de métodos e práticas de engenharia de software como, também, passem a considerar características relacionadas ao ambiente de trabalho, ao conhecimento e experiência das equipes envolvidas e à própria cultura e experiência da organização em desenvolver software [11].

## **2.2 Os Processos Especializados**

Durante a etapa de especialização são produzidos os processos especializados. Estes processos 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. As atividades do processo padrão são detalhadas de acordo com o paradigma e os métodos de desenvolvimento escolhidos de acordo com as características do desenvolvimento de software na organização, tais como experiência dos desenvolvedores e tipos de software desenvolvidos.

Diferentes processos de software podem ser especializados de acordo com as características de cada paradigma de desenvolvimento, podendo ser incluídas novas atividades, porém mantendo os elementos básicos definidos no Processo Padrão. De forma semelhante, pode haver a necessidade de incluir atividades específicas para o tipo de software e/ou de especializar atividades já previstas. Exemplos de paradigma são o estruturado, o orientado a objetos e o baseado em conhecimento. Exemplos de tipos de software são software para Web, software crítico ou para instrumentação virtual.

## **2.3 Os Processos Instanciados**

Este tipo de esforço de padronização sofre com um problema inerente: para acomodar todos os tipos de iniciativas de desenvolvimento em uma organização, o padrão vai inevitavelmente estar em um nível de abstração que atenda às necessidades de todos os projetos, mas não vai ser capaz de fornecer apoio específico às atividades individuais do projeto. A diversidade de projetos de tecnologia da informação frustra qualquer tentativa direta de sistematizar os processos usados para seu desenvolvimento, não existindo um único formato adequado para, efetivamente, apoiar a realização de todos os projetos. É necessária uma abordagem mais flexível e configurável para definição do processo, uma maneira de adaptar o processo às necessidades específicas de cada projeto [9, 12].

A Norma ISO/IEC 12207 [4] reconhece esta necessidade no processo de adaptação ao tratar da implementação do processo de desenvolvimento. Esta Norma é, portanto, flexível para diversas abordagens de engenharia de software, sendo utilizável com qualquer modelo de ciclo de vida, qualquer método ou técnica de engenharia de software e qualquer linguagem de programação. Estas questões são muito dependentes do projeto e do estado da arte da tecnologia. Portanto, estas escolhas são deixadas a critério dos usuários da Norma [13].

Na etapa da instanciação, as características do projeto e da equipe, além da disponibilidade de recursos, são considerados 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 [14, 15]. 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 a execução do projeto.

A adaptação do processo às necessidades específicas de projetos requer um processo pelo qual desenvolvedores e gerentes sejam capazes de instanciar um processo adequado ao projeto em questão. Durante o planejamento de um novo projeto o gerente identifica e analisa

as características principais deste projeto que são usadas como base para a adaptação do padrão gerando um processo instanciado para o projeto. Isso inclui a seleção de atividades e métodos do modelo de processo genérico que devem fazer parte do processo instanciado. Implica, também, na seleção de um modelo de ciclo de vida e no mapeamento das atividades do processo para este modelo [1, 11, 12, 16, 17, 18].

É uma tarefa difícil abstrair o modelo de processo a partir de um padrão e, então, instanciá-lo para um projeto específico. Em uma situação típica, a pessoa responsável pela definição do processo parte do processo padrão e, mentalmente, o mapeia para o modelo de ciclo de vida desejado. Durante ou após a definição deste processo, é feita uma verificação manual do processo, para garantir sua conformidade com o padrão.

Ao tratar da atividade de implementação do processo, a Norma ISO/IEC 12207 [4] sugere:

- Definir ou selecionar um modelo de ciclo de vida de software, apropriado às características do projeto e mapear as atividades e tarefas do processo de desenvolvimento na estrutura deste modelo de ciclo de vida.
- Selecionar, adaptar e utilizar padrões, métodos, ferramentas e linguagens de programação apropriados para executar as atividades do processo de desenvolvimento e de apoio.
- Desenvolver planos para conduzir as atividades do processo de desenvolvimento, incluindo padrões, métodos, ferramentas, ações e responsabilidades associados com o desenvolvimento.

O mapeamento das atividades para um modelo de ciclo de vida engloba [19]:

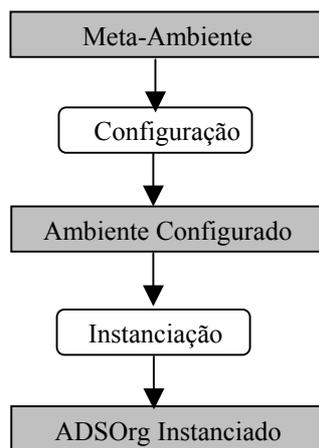
- (i) Comparar atividades com o modelo e identificar quais atividades não são necessárias para o modelo de ciclo de vida e requisitos do projeto. Este passo deve garantir que todas as atividades necessárias ao projeto sejam consideradas.
- (ii) Identificar quantas instâncias de cada atividade são apropriadas. Caso o modelo de ciclo de vida selecionado seja iterativo, uma determinada atividade poderá ser realizada diversas vezes, de acordo com o número de iterações do modelo, enquanto no modelo cascata a mesma atividade será realizada uma única vez.
- (iii) Colocar as atividades em seqüência e checar o fluxo de informações. As dependências entre as atividades devem ser respeitadas para garantir que o processo será realizado corretamente.

### **3. Ferramentas para Definição de Processos**

A estratégia para construção de ADSOrg é iniciada com 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 à 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 [22], 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 2 exibe o esquema atual utilizado para a construção de ADS 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 apoiadas por serviços e ferramentas presentes na Estação TABA. 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 desta estratégia: 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.



**Figura 2 – Esquema para Construção de ADS na Estação TABA**

A sub-seção 3.1 apresenta o processo de configuração de um ambiente para uma organização, tarefa que envolve a definição de processos padrão e especializados para a organização e é realizada através da ferramenta Config. A seção 3.2 apresenta a ferramenta AdaptPro, presente nos Ambientes Configurados, responsável pela definição de processos instanciados para projetos específicos e, também, pela geração dos ADSOrg Instanciados. A sub-seção 3.3 apresenta como o conhecimento de especialistas sobre a definição processos de software pode ser consultado durante a execução destas ferramentas e, também, dos ambientes.

### **3.1 Definição dos processos padrões e especializados**

A definição dos processos padrão e especializados para uma organização é feita durante a configuração de um ambiente próprio para a organização pela ferramenta Config [22].

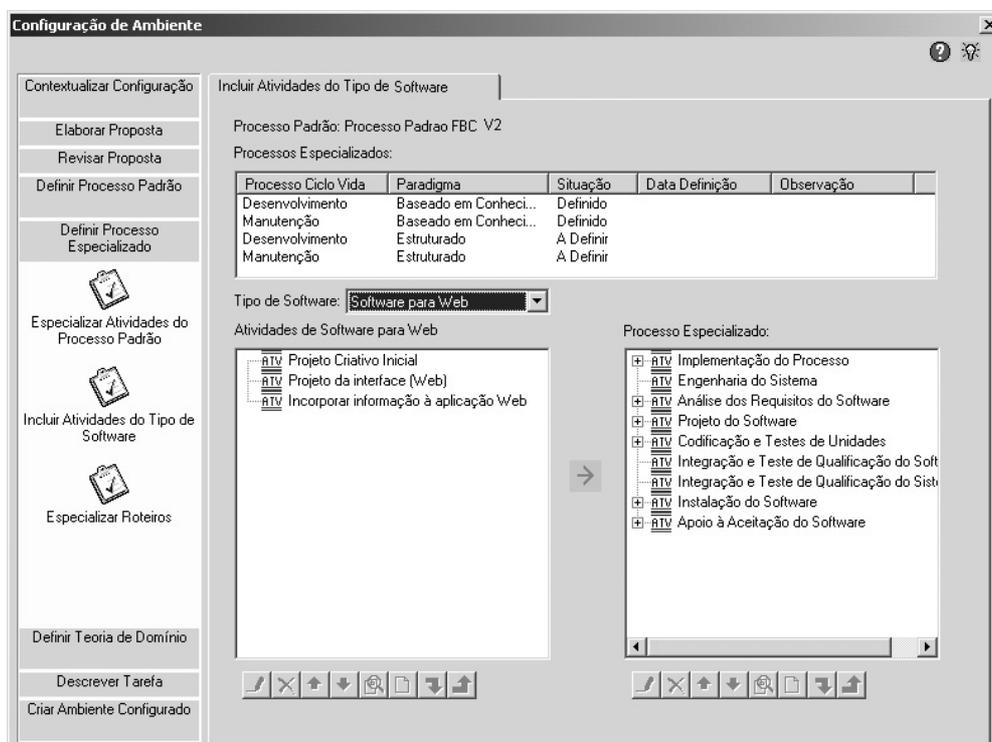
As atividades envolvidas na configuração de um ambiente englobam:

- **Contextualizar a Configuração:** 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. Também são identificados o estágio de adoção de processos de software e de sistemas de gestão da qualidade, práticas adotadas no desenvolvimento, principais problemas no desenvolvimento de software e seus objetivos organizacionais relacionados à melhoria dos processos de software.
- **Elaborar Proposta para Configuração do Ambiente:** 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. Também inclui identificação das teorias de domínio a serem utilizadas, processos a

serem definidos e/ou revistos. A ferramenta também possibilita o registro de pareceres dos Representantes da Organização sobre a proposta de configuração, indicando a sua aprovação ou rejeição e a necessidade de modificações e ajustes.

- **Definir Processo Padrão:** baseado nas características da organização e da forma como o desenvolvimento de software é realizado nesta organização, esta etapa prevê a definição do processo padrão da organização, que pode contemplar os processos de desenvolvimento e de manutenção de software. As etapas envolvidas na definição do processo padrão são:
  - **Caracterizar o processo padrão** a ser definido, estabelecendo quais são os processos do ciclo de vida que devem ser contemplados e se as definições dos processos do ciclo do vida devem ser baseadas ou não em processos definidos previamente;
  - **Incluir as atividades do processo padrão selecionado como base** na definição do novo processo padrão (quando pertinente);
  - **Incluir atividades da Norma 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. Estas atividades não poderão ser retiradas dos processos especializados e instanciados a serem definidos a partir dele;
  - **Adaptar**, para a organização, **os roteiros de documentos** disponíveis no meta-ambiente.
- **Definir Processo Especializado:** definir, a partir do processo padrão, processos especializados para os paradigmas de desenvolvimento utilizados na organização e, opcionalmente, para diferentes tipos de software. As etapas envolvidas na definição dos processos especializados são:
  - **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. A Figura 3 mostra a inclusão de atividades próprias ao desenvolvimento de software para Web.
  - **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.
  - **Definir o mapeamento de atividades para os modelos de ciclo de vida**, ou seja, as atividades do processo especializado devem ser organizadas de acordo com as fases definidas para os modelos de ciclo de vida. Para isto deve-se (i) verificar a existência no repositório da organização de um mapeamento feito para um projeto anterior, considerando o mesmo processo padrão e o mesmo modelo de ciclo de vida; ou (ii) realizar o mapeamento caso não tenha sido

encontrado um mapeamento anterior. Este mapeamento será utilizado durante a instanciação dos processos de software, a ser descrita na próxima sub-seção.



**Figura 3 – Tela de Inclusão de Atividades do Tipo de Software**

- **Definir Teoria do Domínio e Descrever Tarefa:** definir uma Teoria do Domínio para um determinado domínio de conhecimento e descrever uma tarefa genérica que tenha sido identificada como pertinente ao contexto de uma subteoria do domínio.
- **Criar Ambiente Configurado:** criar o Ambiente Configurado para a organização.

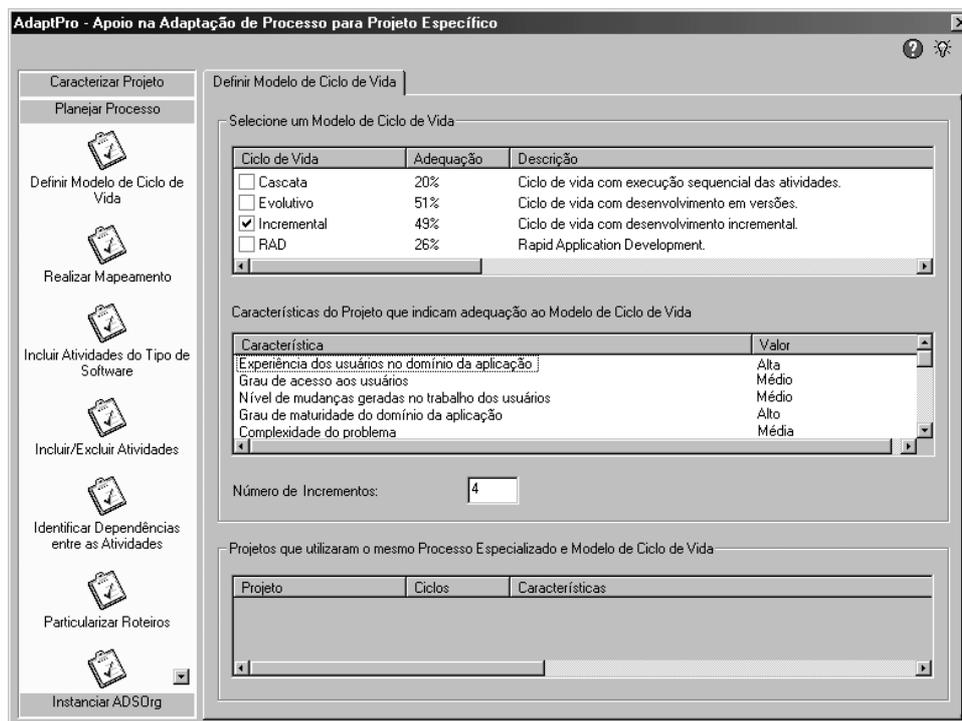
O ambiente configurado para a organização irá conter o processo padrão e os processos especializados além de conhecimento específico sobre o desenvolvimento e manutenção de software. A partir deste ambiente a organização será capaz de gerar ambientes instanciados para cada projeto a ser desenvolvido. Para esta tarefa, foi definido um processo de instanciação de ADSOrg [14] que será apresentado a seguir.

### 3.2 Definição dos processos instanciados

Para ser utilizado em um projeto específico, o processo especializado mais adequado deve ser instanciado para atender às características do projeto. Esta sub-seção apresenta uma abordagem para a implementação do processo de desenvolvimento de software em um ambiente configurado através da utilização da ferramenta AdaptPro [14], cujo objetivo é apoiar o gerente de projetos durante a adaptação do processo, utilizando técnicas de gerência do conhecimento. Foi definido para isso um processo de instanciação de processos de software, tornando explícito o conhecimento acumulado em diversos projetos, cujo resultado da instanciação é um processo instanciado para um projeto específico. As etapas envolvidas na instanciação são:

- **Caracterizar Projeto:** esta atividade visa identificar as características do projeto para o qual será realizada a instanciação, obtendo informações que servirão como base para a tomada de decisões.

- **Definir o projeto** consiste em definir as características do novo projeto a ser realizado na organização para a qual será instanciado o processo. São especificados, por exemplo, que tipos de software serão desenvolvidos, os paradigmas a serem utilizados e a natureza do projeto (desenvolvimento de um novo software ou manutenção). Também são identificados as organizações desenvolvedoras e clientes do projeto.
- **Identificar as características do projeto** em relação ao software a ser desenvolvido a partir de informações sobre o seu escopo, tamanho, complexidade, tipo, domínio da aplicação, conhecimento da equipe sobre o domínio da aplicação, tecnologia de desenvolvimento, disponibilidade de recursos financeiros e de tempo, requisitos de qualidade e criticidade do software. Três tipos de características são coletadas:
  - *características gerais* que servem de apoio para a identificação do modelo de ciclo de vida mais adequado ao projeto e também para identificação de projetos similares. A lista das características e informações sobre o mapeamento realizado podem ser encontrado em [14];
  - *características de qualidade definidas na Norma ISO/IEC 9126* [24] para produtos de software. As seis características definidas podem não estar todas presentes em um produto ou, então, não serem necessárias no mesmo grau. Por isso é necessário determinar que características devem estar presentes no produto, e em que grau, através de avaliações das pessoas envolvidas no projeto. Os avaliadores atribuem para cada característica de qualidade o grau em que ela é necessária na aplicação a ser desenvolvida. Para se chegar a este grau de relevância geral, a partir do julgamento individual, é utilizada a lógica *fuzzy* através da ferramenta QFuzzy [1].
  - *Nível de garantia da qualidade necessário* considerando-se o dano causado por falhas [20].
- **Planejar Processo:** Esta atividade tem como objetivo a geração do processo instanciado para o novo projeto. Quando um ambiente é configurado, são fornecidos o processo padrão e os processos especializados para as tecnologias de desenvolvimento utilizadas na organização. O processo especializado mais adequado ao projeto é selecionado de acordo com o paradigma de desenvolvimento. O processo de instanciação parte deste processo especializado e adapta-o para adequá-lo às características específicas do projeto. As atividades obrigatórias, definidas durante a configuração do ambiente para a organização, não poderão ser excluídas do processo durante esta etapa.
  - **Definir o modelo de ciclo de vida** consiste em selecionar o modelo de ciclo de vida mais adequado ao projeto (Figura 4). Com base nas características do projeto, são oferecidas ao gerente do projeto as opções de modelo de ciclo de vida mais apropriadas ao projeto. O gerente do projeto deve, então, selecionar o modelo de ciclo de vida que julgar mais adequado baseado nos critérios definidos na ferramenta ou devido a sua experiência pessoal. Um modelo de ciclo de vida define um série de etapas a serem realizadas para o desenvolvimento de um produto de software.



**Figura 4 – Escolha do Modelo de Ciclo de Vida a ser Adotado no Projeto**

As atividades do processo são organizadas de acordo com essas etapas, que podem ser estruturadas de maneira seqüencial e, nesse caso, são realizadas apenas uma vez, ou de maneira iterativa, sendo realizadas várias vezes. Neste momento deve-se definir as atividades que serão realizadas em cada uma dessas etapas e, quando apropriado, o número de iterações. Caso o modelo de ciclo de vida seja iterativo, o gerente de projeto deve definir o número de iterações a serem realizadas (no caso do modelo evolutivo, o número de ciclos de desenvolvimento; no caso do modelo incremental, o número de incrementos; no caso do modelo RAD, o número de módulos a serem desenvolvidos em paralelo). Para apoiar essa definição, o gerente pode consultar o número de iterações realizadas em projetos anteriores que utilizaram o mesmo processo especializado e modelo de ciclo de vida. A partir do conhecimento sobre a estrutura de cada modelo de ciclo de vida, a ferramenta define as atividades que compõem cada etapa, realizando um mapeamento inicial, que poderá ser modificado pelo gerente do projeto. Caso o projeto envolva a manutenção de um software, um modelo de ciclo de vida não precisará ser definido.

- **Mapear Atividades do Modelo de Ciclo de Vida** consiste organizar as atividades previstas no processo especializado de acordo com o modelo de ciclo de vida selecionado. Cada modelo de ciclo de vida propõe uma maneira diferente de estruturar as etapas do desenvolvimento:
  - No *modelo cascata* as atividades do processo são executadas em uma única etapa seqüencial.
  - No *modelo incremental* há uma etapa inicial seqüencial, na qual pode ser realizada a especificação de requisitos e o projeto da arquitetura. Os requisitos são, então, segmentados e alocados a diferentes versões. Em seguida há uma etapa iterativa, onde cada iteração corresponde a um incremento, ou seja, uma versão operacional do software que implementa

um subconjunto dos requisitos, de acordo com a definição feita na etapa inicial. Ao final do último incremento, todos os requisitos do software estarão implementados.

- No *modelo RAD* há uma etapa inicial seqüencial, semelhante ao modelo incremental, na qual são definidos os módulos que compõem o produto. A etapa seguinte será realizada de forma paralela para cada um dos módulos do software. Uma equipe é alocada para cada módulo e o desenvolvimento de cada um deles é realizado de acordo com as atividades previstas no processo. Ao final do desenvolvimento dos módulos é realizada uma nova etapa seqüencial para integração dos módulos.
- Caso o projeto seja desenvolvido segundo o paradigma orientado a objetos, é assumido que o processo utiliza a estrutura definida pelo *RUP* [23]. Nesse caso, as atividades do processo são estruturadas em quatro fases: Construção, Elaboração, Construção, Transição.

As atividades do processo especializado são automaticamente distribuídas nas fases e iterações definidas pelo modelo de ciclo de vida, entretanto, o gerente do projeto pode fazer alterações, como:

- ***Incluir atividades específicas do tipo de software a ser desenvolvido*** caso estas não tenham sido incluídas durante a definição do processo padrão ou do processo especializado. Em um ambiente configurado, duas situações podem ocorrer: (i) a organização desenvolve apenas um tipo de software, então, o conhecimento sobre as atividades específicas para este tipo de software já foi incorporado ao processo padrão da organização, uma vez que tais atividades devem ser realizadas em qualquer projeto dessa organização; ou (ii) a organização desenvolve diferentes tipos de software, então, o conhecimento sobre as atividades específicas para cada tipo de software não será incorporado aos processos padrão e especializados, mas estará disponível no ambiente configurado e será incorporado ao processo instanciado para um projeto neste momento. Para apoiar essa atividade, o gerente pode consultar processos instanciados de projetos anteriores que desenvolveram o mesmo tipo de software e utilizaram o mesmo processo especializado para verificar as atividades que foram inseridas no processo e em que ordem.
  - ***Incluir/excluir atividades*** do processo caso não deseje realizar todas as presentes no processo especializado. Entretanto, as atividades definidas como sendo obrigatórias durante o processo de configuração do ambiente não poderão ser retiradas. Durante esta atividade a ordem de execução das atividades no processo também pode ser modificada.
  - ***Identificar dependências entre as atividades do processo*** definindo quais são as pré-atividades de cada atividade.
  - ***Particularizar roteiros de documentos*** definidos para a organização quando necessário e de acordo com as características do projeto a ser desenvolvido.
  - ***Avaliar o processo instanciado*** para determinar se ele atende os objetivos definidos. Esta avaliação pode envolver tanto a equipe responsável pela execução do projeto quanto os contratantes. Caso o processo seja considerado inadequado ele poderá ser modificado e, então, reavaliado.
- **Instanciar ADSOrg:** uma vez que o processo de desenvolvimento esteja definido e aprovado é possível instanciar um ambiente para o projeto. A partir deste ambiente, o processo poderá ser acompanhado e os artefatos definidos poderão ser gerados a partir

das ferramentas disponíveis nos ADSOrg ou de ferramentas externas integradas aos mesmos.

### 3.3 Acesso ao Conhecimento durante a definição de Processos

Todas as ferramentas construídas no contexto do projeto Ambientes de Desenvolvimento Orientados a Organização utilizadas para apoiar atividades dos processos de desenvolvimento e manutenção de software, incluindo o *Config* e o *AdaptPro*, são orientadas a processo e integradas à ferramenta *Acknowledge* [21]. Todas as telas destas ferramentas, como pode ser visto nas Figuras 3 e 4, possuem no lado direito superior da tela, dois ícones cuja finalidade é fornecer o conhecimento disponível no ambiente para auxiliar a execução da atividade (?) e permitir o registro de conhecimento obtido na execução da atividade (?). Isto significa que as ferramentas possibilitam a gerência de conhecimento orientada a processo ao fornecer conhecimento que é relevante para a atividade que está sendo executada e ao permitir o registro do conhecimento adquirido na execução da atividade.

Durante a definição dos processos instanciados, por exemplo, não é possível garantir que o gerente de projeto responsável tenha experiência na definição de processos, ou mesmo em engenharia de software. Portanto, torna-se necessário oferecer meios de apoiar a atividade de instanciação, para garantir a qualidade do processo gerado. Por exemplo, caso o gerente deseje obter ajuda sobre a realização da atividade de mapeamento das atividades de um processo para um determinado modelo de ciclo de vida, como pode ser visto na Figura 4, bastar clicar sobre o ícone correspondente e os diferentes tipos de conhecimento disponíveis são listados. A Figura 5 exibe como uma lição aprendida sobre o uso do modelo de ciclo de vida evolutivo é disponibilizada. O gerente do projeto pode, também, caso deseje, avaliar o grau de utilidade da lição aprendida, que será utilizado para apoiar a atividade de manutenção do conhecimento armazenado no repositório da organização.

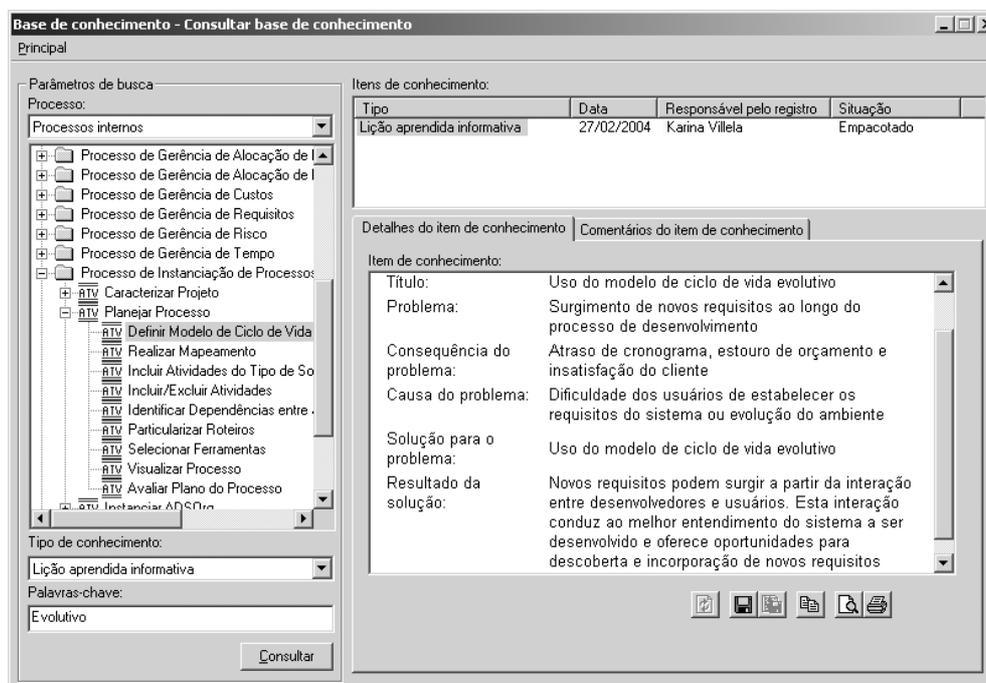


Figura 5 – Busca e Acesso do Conhecimento Desejado

#### **4. Considerações Finais**

Este artigo apresentou a abordagem adotada pela Estação TABA para a definição de processos de software para organizações e projetos específicos dentro do contexto dos Ambientes de Desenvolvimento de Software Orientados a Organização. Esta abordagem engloba a definição de processos em três níveis: padrão, especializado e instanciado. Também foram apresentadas as ferramentas Config e AdaptPro construídas para dar suporte a esta abordagem.

A abordagem descrita, 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. Também promove o uso deste conhecimento na definição de processos, apoiando a Gerência do Conhecimento em Ambientes de Desenvolvimento de Software.

Desde o final de 2003 os ADSOrg começaram a serem utilizados por 18 pequenas e médias empresas de software brasileiras. Duas grandes empresas públicas brasileiras de atuação nacional estarão iniciando sua utilização em 2004. Resultados iniciais são promissores, porém ainda é cedo para uma avaliação completa. O conjunto atual de usuários já nos permite uma avaliação das ferramentas através de um estudo experimental que está sendo planejado e será executado ao longo deste ano.

#### **Agradecimentos**

Agradecemos ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e pelo apoio financeiro ao projeto Ambientes de Desenvolvimento de Software Orientados à Organização e a todos os colaboradores deste projeto.

#### **Referências Bibliográficas**

- [1] 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.
- [2] VILLELA, K., OLIVEIRA, K. M., SANTOS, G., ROCHA, A. R. C., TRAVASSOS, G. H. Cordis-FBC: an Enterprise Oriented Software Development Environment In: Workshop Learning Software Organization, Luzern, 2003
- [3] 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.
- [4] ISO/IEC, ISO/IEC 12207: Information Technology – Software Life Cycle Process, International Organization for Standardization and International Electrotechnical Commission, Geneva, Switzerland, 1995.
- [5] 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.
- [6] 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.
- [7] SEI, Capability Maturity Model Integration (CMMI) Version 1.1 - Staged Representation, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, 2002.

- [8] MAURER, F., DELLEN, B., HOLZ, H., 1999, Process Support for Virtual Software Organizations, First International Workshop on Learning Software Organizations (LSO 1999), pp.87-98.
- [9] HENNINGER, S., 2001, "Turning Development Standards into Repositories of Experiences", Software Process Improvement and Practice 2001, 6: 141-155.
- [10] EMAM, K. E., DROUIN, J., MELO W., SPICE – The Theory and Practice of Software Process Improvement and Capability Determination, IEEE Computer Society Press,1998.
- [11] MACHADO, L.F.D., 2000, Modelo para Definição de Processos de Software, Tese de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, Março.
- [12] HENNINGER, S., 1999, Using Software Process to Support Learning Software Organizations, First International Workshop on Learning Software Organizations (LSO 1999), pp.99-114.
- [13] ROCHA, A.R. C., MALDONADO, J.C., WEBER, K.C., 2001. Qualidade de Software. São Paulo. Prentice Hall.
- [14] BERGER, P., Instanciação de Processos de Software em Ambientes Configurados na Estação TABA, Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2003.
- [15] 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.
- [16] MAURER, F., DELLEN, B., HOLZ, H., 1999, Process Support for Virtual Software Organizations, First International Workshop on Learning Software Organizations (LSO 1999), pp.87-98.
- [17] HOLZ, H., KÖNNECKER, A., MAURER, F., 2001, Task-Specific Knowledge Management in a Process-Centred SEE. In: K.-D. Althoff, R.L. Feldmann, W. Müller (Eds.): LSO 2001, LNCS 2176, pp. 163-177. Springer-Verlag Berlin Heidelberg 2001.
- [18] BORGES, L. M. S., 2001, Uma Ferramenta para Instanciação de Processos de Software e Apoio ao Compartilhamento de Experiências, Tese de Mestrado, UFES, Vitória, ES, Brasil, Dezembro.
- [19] CRISTENSEN, M. J., THAYER, R. H., The Project Manager's Guide to Software Engineering Best Practices, IEEE Computer Society.
- [20] BOEGH, J., HAUSEN, H-L E WELZEL, D.; "A Practitioners Guide to Evaluation of Software"; IEEE Software; 1993
- [21] MONTONI, M., MIRANDA, R., ROCHA, A. R.. TRAVASSOS, G. H., "Knowledge Acquisition and Communities of Practice: An Approach to Convert Individual Knowledge into Multi-Organizational Knowledge", In: Workshop Learning Software Organization, Banff, 2004.
- [22] VILLELA, K., Definição e Construção de Ambientes de Desenvolvimento de Software Orientados a Organização, Tese de D. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, 2004.
- [23] KRUCHTEN, P., The Rational Unified Process – An Introduction, Addison-Wesley.
- [24] ISO/IEC 9126/NBR 13596, 1996, Tecnologia de Informação – Avaliação de Produto de Software – Características de Qualidade e Diretrizes para o seu uso, ABNT – Associação Brasileira de Normas Técnicas.