

# Gerência de Conhecimento sobre Processos de Software

Ligia da Motta Silveira Borges<sup>1,2</sup>  
Ricardo de Almeida Falbo<sup>1</sup>

<sup>1</sup> Mestrado em Informática - UFES  
Av. Fernando Ferrari s/n.º  
CEP 29000-060, Vitória - ES

<sup>2</sup> CDSV SA  
Av. Fernando Ferrari, 1000  
Goiabeiras, CEP 29060-410  
Vitória – ES

Email: ligias@zaz.com.br, falbo@inf.ufes.br

## Resumo

Neste artigo, discute-se a importância de armazenar e compartilhar a experiência obtida no desenvolvimento de software para a melhoria contínua da qualidade e produtividade. Para compartilhar este conhecimento, uma base de experiência deve ser criada contendo o processo padrão da organização, assim como os artefatos e os conhecimentos informais obtidos durante os projetos. Para facilitar o armazenamento e o compartilhamento da experiência, foi desenvolvida uma ferramenta que provê suporte à atividade de adaptação do processo padrão, fornecendo, também, um mecanismo de busca.

**Palavras-chave:** Definição e Melhoria de Processo, Ferramenta de Controle da Qualidade, Processo Padrão, CMM, Gerência de Conhecimento.

## Abstract

In this paper, we discuss the crucial importance of storing and sharing the experience obtained through software development to the continuous improvement of quality and productivity. To share this knowledge, an experience base should be built containing the organizational standard process as well as the artifacts and informal knowledge obtained throughout the projects. In order to facilitate the storage and sharing of the experience, we built a tool that supports the standard process tailoring procedure for each project, providing a search mechanism.

**Keywords:** Process Improvement, Process Definition, Quality Control Tool, Standard Software Process, CMM, Knowledge Management.

## 1 Introdução

A demanda no desenvolvimento de software vem aumentando. Com isso, menores tempos de mercado, maior qualidade e maior produtividade são, cada vez mais, fatores críticos para a competitividade das organizações de software. Ao mesmo tempo, os produtos de software a serem desenvolvidos são também mais complexos.

Essa situação vem motivando a comunidade de engenharia de software a desenvolver pesquisas buscando garantir a qualidade de software, mantendo (ou até mesmo aumentando) a produtividade no desenvolvimento e diminuindo o tempo de resposta ao mercado. Tais pesquisas têm indicado que a qualidade do produto de software é fortemente dependente da qualidade do processo de desenvolvimento.

Neste contexto situa-se o Centro de Desenvolvimento de Sistemas de Vitória (CDSV), um dos cinco centros de desenvolvimento de software da Xerox no mundo. O CDSV tem investido na qualidade de seu processo de desenvolvimento, tendo sido certificado como nível de maturidade 3 de acordo com o CMM [1].

Contudo, para ser eficaz e conduzir à construção de produtos de boa qualidade, um processo deve ser adequado ao domínio da aplicação e ao projeto específico. Desse modo, processos devem ser definidos caso a caso considerando-se as especificidades da aplicação, a tecnologia a ser adotada na sua construção e o grupo de desenvolvimento [2]. Sendo assim, a adaptação do processo padrão para cada projeto passa a ser uma tarefa crucial que, infelizmente, não é simples. Portanto, oferecer apoio automatizado a sua realização é um importante desafio.

Este trabalho apresenta uma ferramenta desenvolvida para apoiar a manutenção e melhoria contínua da qualidade do processo, apoiando a adaptação do processo padrão do CDSV para os projetos e permitindo o compartilhamento de conhecimentos adquiridos nesses projetos. A gerência do conhecimento de processos de software visa oferecer meios de apoiar a melhoria contínua do processo, assim como permitir o aprendizado da organização como um todo.

Na seção 2, discute-se o processo de desenvolvimento de software e as dificuldades encontradas na sua adaptação. A seção 3 relata sucintamente os esforços do CDSV na busca da melhoria contínua do processo. A seção 4 trata da gerência de conhecimento como apoio à melhoria contínua do processo. A seção 5 apresenta ProKnowHow, uma ferramenta de apoio à melhoria de processos. Na seção 6, são discutidos trabalhos correlatos. Finalmente, a seção 7 apresenta as conclusões deste trabalho.

## **2 Processo de Desenvolvimento de Software**

Um processo de software pode ser visto como um conjunto de atividades, métodos, ferramentas e práticas que são utilizadas para construir um produto de software. Humphrey [3] define processo de software como o conjunto de tarefas de engenharia de software necessárias para transformar os requisitos dos usuários em software. Na definição de um processo de software devem ser consideradas as seguintes informações: atividades a serem realizadas, recursos necessários, artefatos requeridos e produzidos, procedimentos adotados e o modelo de ciclo de vida utilizado [2].

No entanto, não existe um processo de software que possa ser genericamente aplicado a quaisquer projetos, visto que nenhum projeto é idêntico ao outro. Variações na tecnologia e paradigma adotados no desenvolvimento, tamanho e complexidade do projeto, requisitos e métodos de desenvolvimento, entre outros fatores, influenciam na forma como um produto de software é adquirido, desenvolvido, operado e mantido [4]. Assim, na definição de um processo, deve-se considerar a sua adequação às tecnologias envolvidas, ao tipo de software em questão, ao domínio de aplicação, ao grau de maturidade (ou capacitação) da equipe em engenharia de software, às características próprias da organização e às características do projeto e da equipe [5].

Em uma organização, diversos projetos podem coexistir possuindo características específicas. Porém, existe um conjunto de elementos fundamentais que devem ser incorporados a quaisquer processos definidos. Esse conjunto de elementos fundamentais é chamado de processo padrão, ou seja, o processo básico que guia o estabelecimento de um processo comum na organização. Desta forma, um processo padrão define uma estrutura única a ser seguida por todas as equipes envolvidas em projetos de software, independentemente das características do software a ser desenvolvido. Humphrey define um conjunto de razões para a definição de um processo padrão [3]:

- Redução dos problemas relacionados a treinamento, revisões e suporte a ferramentas;
- As experiências adquiridas nos projetos são incorporadas ao processo padrão e contribuem para melhorias em todos os processos definidos;
- Economia de tempo e esforço na definição de novos processos adequados a projetos.

Na literatura atual, observa-se uma tendência em se utilizar o processo padrão para definir os processos. As normas ISO/IEC 12207 [4], ISO/IEC TR 15504 [6] e o Modelo de Maturidade e Capacidade (*Capability Maturity Model – CMM*) [7] definem um processo padrão como um ponto base a partir do qual um processo especializado poderá ser obtido de acordo com as características de um projeto de software específico. Desta forma, ser adaptável e configurável torna-se um importante requisito a ser atingido na definição de um processo padrão. Além disso, é importante que esse processo evolua continuamente, para que, cada vez mais, a qualidade seja aperfeiçoada.

### **3 Melhoria Contínua de Processos no CDSV**

Em 1998, ao ser certificado como nível 3 de maturidade do CMM, o CDSV adotou o estabelecimento de um projeto anual de melhoria do processo, o Projeto SPI (*Software Process Improvement*), como abordagem para a melhoria contínua. Assim, a cada ano são estabelecidas metas e, tomando como base essas metas, é feito o planejamento das ações de melhoria.

Com base nos projetos SPI já realizados, os seguintes pontos foram apontados como os mais problemáticos nos esforços de melhoria do CDSV:

- Instanciação do processo para projetos específicos;
- Divulgação do conhecimento adquirido durante os projetos;
- Melhoria contínua do processo de software através de realimentação.

É importante que haja um procedimento que permita a alteração do processo padrão da organização, propiciando a melhoria contínua do mesmo, principalmente no que se refere aos novos procedimentos que ainda estão imaturos e, fatalmente, precisam ser alterados à medida que vão sendo utilizados.

Mas não basta somente atualizar o processo da organização, é necessário que as lições aprendidas durante os projetos de desenvolvimento sejam disseminadas por toda a organização. Portanto, uma organização que não registra os detalhes que envolvem o sucesso ou o fracasso de seus projetos terá como resultado a repetição das falhas. Esse aprendizado da organização como um todo, através dos acertos e falhas nos projetos, tem se mostrado ainda pouco eficiente no CDSV. Para solucionar esse problema, é necessário investir tempo e esforço para estabelecer procedimentos sistemáticos, capazes de propiciar que não somente um pequeno grupo, mas a organização como um todo aprenda através dos erros.

Tendo em vista esse quadro, optou-se por tentar oferecer apoio automatizado a essas tarefas. Contudo, fazendo-se uma análise mais crítica, foi possível constatar que o principal problema está, de fato, na falta de compartilhamento do conhecimento sobre processos adquirido nos vários projetos.

#### **4 Gerência de Conhecimento do Processo**

Para apoiar a instanciação do processo de software e a divulgação das lições aprendidas durante os projetos e garantir uma melhoria contínua do processo, existem algumas tecnologias que podem ser utilizadas, dentre elas a gerência de conhecimento.

O conhecimento tem sido considerado como o patrimônio mais importante de uma empresa, possuindo influência decisiva em sua competitividade. No entanto, para a vantagem competitiva fornecida pelo conhecimento ser sustentável, o conhecimento não pode estar no nível de indivíduo. O conhecimento organizacional registrado em papel também representa um problema, pois não pode ser facilmente acessado, compartilhado e atualizado [8, 9]. Dessa forma, as organizações têm adotado a gerência de conhecimento como forma de melhorar o fluxo de conhecimento entre seus funcionários, capturar o conhecimento organizacional e promover o surgimento de conhecimento novo.

O termo gerência do conhecimento surgiu nos anos 90 como consequência do reconhecimento da importância do conhecimento para a sobrevivência e o sucesso das organizações. A gerência de conhecimento consiste em coletar e armazenar sistematicamente o conhecimento adquirido e compartilhar esse conhecimento através de uma memória corporativa.

No caso do foco na melhoria do processo, podemos, também, chamar esse conhecimento de experiência, o que vai de encontro com o conceito de fábrica de experiência [11], uma abordagem para melhoria da qualidade de software, que tem motivação semelhante a da gerência de conhecimento, porém se concentrando na melhoria contínua de qualidade em organizações que desenvolvem e mantêm software.

Assim, para a melhoria contínua da qualidade e produtividade no desenvolvimento de software, as organizações devem construir uma base de experiências em engenharia de software [12]. Para atingir essa meta, deve-se realizar, para cada projeto, as seguintes atividades [13,14]:

- Caracterizar o projeto corrente;
- Escolher o modelo de processo, os métodos e as ferramentas de apoio apropriados para o projeto, definindo o plano do projeto;
- Executar o processo, construindo produtos e coletando e analisando dados para fornecer realimentação (*feedback*) em tempo real;
- Analisar os dados para avaliar as práticas correntes, determinar problemas, registrar descobertas e recomendar melhorias para projetos futuros;
- Empacotar a experiência na forma de modelos refinados e atualizados e em outras formas de conhecimento estruturado, registrando-a na Base de Experiências para reuso em futuros projetos.

A interação entre organização do projeto e fábrica de experiências estabelece dois ciclos de realimentação. O primeiro ocorre no passo de execução do processo, permitindo que a experiência obtida durante o processo seja analisada e pequenas mudanças sejam realizadas, o que significa aprendizado em nível de projeto. O segundo permite o empacotamento da experiência ao final do projeto e a utilização dessa experiência em um novo projeto,

resultando em aprendizado corporativo [13, 14]. Nesse contexto, surge uma nova função na organização, o Gerente de Conhecimento, que centraliza a gerência dos itens de conhecimento e as iniciativas de aprendizado organizacional [9, 14].

As metas organizacionais determinam o tipo de conhecimento a ser coletado por uma organização. Qualquer tipo de experiência pode ser reutilizado com intuito de evitar retrabalho e melhorar a qualidade, entre eles, modelos, especificações, código, materiais de treinamento, melhores práticas em engenharia do software, lições aprendidas, etc [13, 15]. Além disso, o armazenamento adequado dos itens de experiência também deve ser considerado em uma abordagem eficiente de reuso. Desta forma, o item de experiência gerado em um projeto deve ser adaptado para reuso em projetos futuros, compondo, junto com informações adicionais capazes de facilitar o reuso, um pacote de experiência.

## **5 ProKnowHow: uma Ferramenta de Apoio à Melhoria de Processos**

Para ajudar uma organização na busca da melhoria contínua do processo de software, é útil fornecer apoio automatizado por meio de uma ferramenta capaz de suportar o armazenamento e disseminação de experiências. Essa ferramenta deve possuir um repositório contendo os conhecimentos formal e informal obtidos nos projetos, organizados de forma a poderem ser reutilizados em outros projetos.

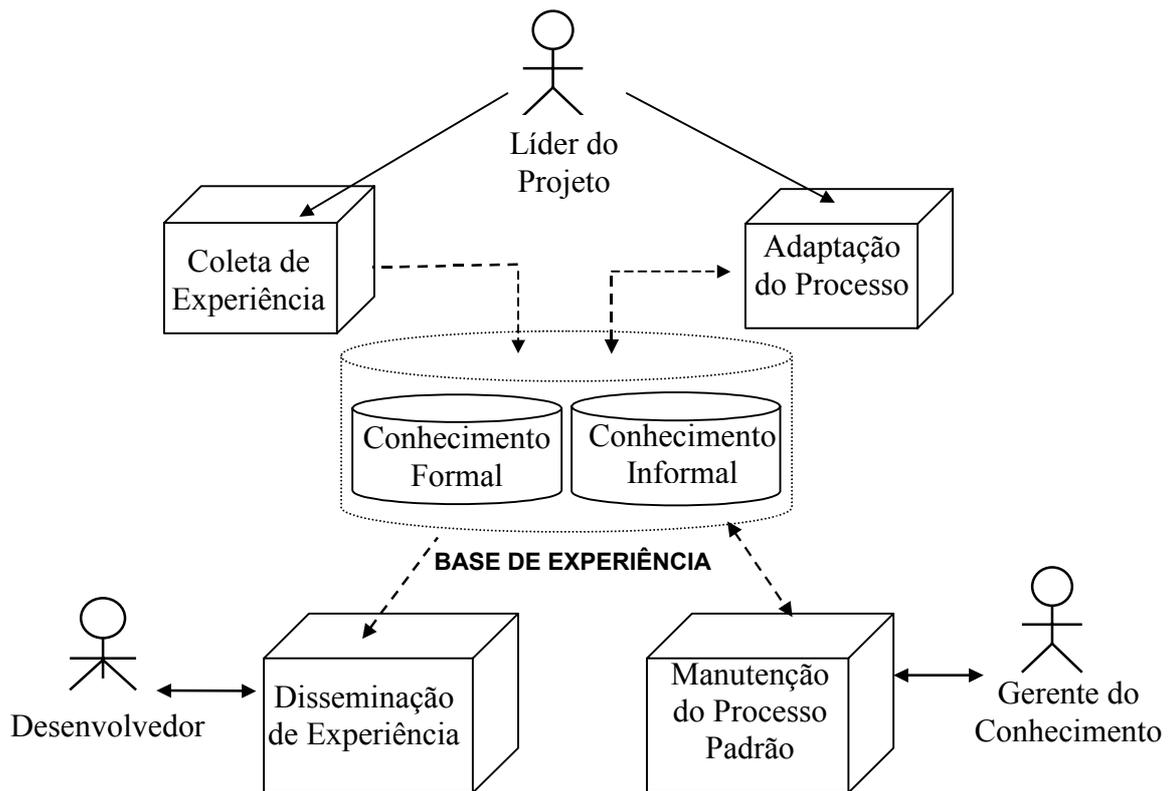
A ferramenta ProKnowHow foi desenvolvida com o objetivo principal de apoiar a disseminação do conhecimento sobre processos. Contudo, são também seus objetivos:

- Apoiar a instanciação do processo para projetos específicos;
- Divulgar o conhecimento adquirido durante os projetos;
- Apoiar a melhoria contínua do processo de software através da realimentação.

Em ProKnowHow, o conhecimento é armazenado em uma base de experiência no formato de lições aprendidas com o seguinte esquema de caracterização:

1. Projeto - projeto a qual pertence;
2. Objeto - item do processo ao qual a lição está associada, tais como atividade, artefato, procedimento, ferramenta, etc;
3. Tipo de lição aprendida - identifica se essa lição foi um ponto positivo (boa prática) ou um ponto negativo (oportunidade de melhoria);
4. Categoria de problema - categorias fixas pré-definidas de problemas, tais como gerência de projeto, planejamento de projeto, gerência de requisitos, etc;
5. Problema - descrição do problema;
6. Solução - descrição da solução adotada para o problema;
7. Contexto - descrição da situação onde o problema/solução é relevante.

É importante realçar que a construção da base de experiência foi apoiada em uma ontologia de processo de software [2]. Essa ontologia foi utilizada para tornar homogêneo o vocabulário, facilitar o compartilhamento do conhecimento sobre processos e simplificar a busca de experiências. Assim, a descrição dos objetos da base de experiência (item 2) é feita usando os termos da ontologia e buscas feitas com sinônimos são transformadas para o vocabulário da ontologia. Para alcançar seus objetivos, a ferramenta possui a arquitetura apresentada na Figura 1.

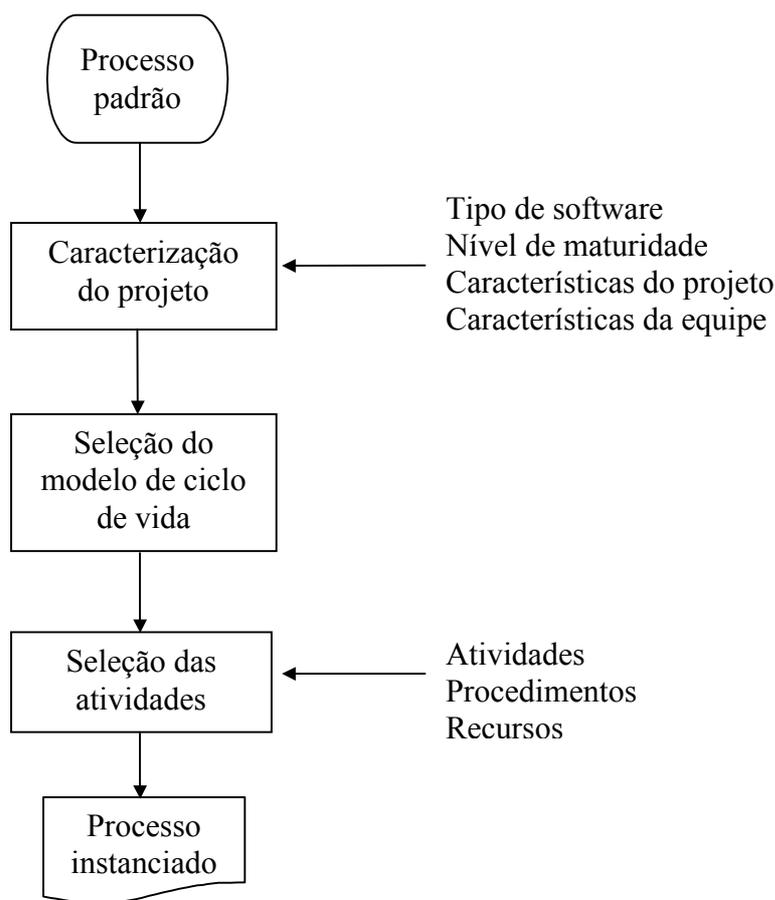


**Figura 1- Arquitetura de ProKnowHow**

ProKnowHow contém um repositório para o processo padrão da organização que é composto pelos modelos de ciclo de vida, procedimentos, atividades, ferramentas, modelos de documentos e políticas. Toda a infraestrutura necessária à manutenção do processo e dos seus componentes também é fornecida. Essa manutenção é efetuada de acordo com as regras estabelecidas pela organização, que define como a necessidade de alteração no processo é detectada, registrada, assim como os recursos para efetuar essa alteração e os procedimentos para a aprovação do mesmo.

### 5.1 - Adaptação do Processo

A ferramenta apóia o líder do projeto na adaptação do processo padrão para cada projeto, apresentando sugestões, tanto de modelo de ciclo de vida, quanto de atividades, procedimentos e ferramentas, entre outros. No entanto, o líder é livre para acatar, ou não, as sugestões fornecidas pela ferramenta. Porém, para garantir a qualidade da adaptação, o gerente deve realimentar o sistema com seus comentários, que podem ser críticas ou justificativas, sobre a orientação que recebeu. Sabe-se que, à medida que o produto de software vai sendo desenvolvido, uma visão mais precisa das necessidades reais do projeto, sejam ferramentas, documentos ou mesmo treinamentos, fica mais precisa. Portanto, oferece-se ao gerente, também, um meio para alterar o processo do projeto ao longo do desenvolvimento do produto. A figura 2 mostra um esquema do modelo utilizado pelo CDSV na adaptação do processo padrão, composto por três atividades principais: caracterização do projeto, seleção do ciclo de vida e seleção das atividades.



**Figura 2- Modelo de adaptação do processo padrão**

Na fase de caracterização do projeto, as características do problema, do projeto, da equipe, assim como o nível de maturidade desejado para o projeto, são informados, como mostra a Figura 3. O objetivo principal é incorporar ao processo instanciado, características específicas do projeto. As seguintes informações são fundamentais:

- Tipo de Software - o tipo de software a ser desenvolvido tem impacto no processo de software a ser definido. Sistemas para a Web, por exemplo, requerem que o processo de desenvolvimento inclua atividades que tratem adequadamente a interação homem-máquina. No CDSV, os produtos de software podem ser classificados nos seguintes tipos: (i) Sistemas de tempo-real; (ii) Sistemas de informação; (iii) Software básico e (iv) Sistema para Web. Esta classificação foi baseada no histórico de produtos desenvolvidos pela organização, assim como prospecções de seus rumos.
- Nível de Maturidade - o nível de maturidade deve ser selecionado para se saber que atividades do processo padrão considerar. É possível selecionar um nível fixo de maturidade, assim, como personalizar o nível de maturidade escolhendo quais áreas-chave devem ser atingidas naquele projeto.

- Características do projeto - foram adotadas as características propostas em [5] e os seguintes critérios específicos do CDSV: (i) tipo de produto; (ii) tipo de serviço e (iii) sub-contratação ou não de serviços.



**Figura 3 – Caracterização do projeto**

Na definição de um processo de software para um projeto, deve-se considerar a seleção de um modelo de ciclo de vida. São inúmeros os modelos de ciclo de vida propostos na literatura e uma seleção inadequada pode influenciar no sucesso de um projeto. Neste trabalho, foram considerados apenas os modelos de ciclo de vida aprovados para uso no CDSV, sendo que, para a seleção, utilizou-se a sistemática definida em [5].

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

As atividades de gerência são aquelas relacionadas ao planejamento e acompanhamento gerencial do projeto e são selecionadas de acordo com o nível de qualidade e/ou áreas-chave escolhidas para o projeto. As atividades de construção estão diretamente relacionadas ao processo de construção do software e devem ser realizadas mesmo que o projeto não tenha como objetivo atingir as áreas-chave de processo (KPAs) correspondentes.

Sua seleção é, portanto, muito relacionada às características do projeto, assim como do modelo de ciclo de vida selecionado.

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

## **5.2 Coleta de Experiência**

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

ProKnowHow fornece meios de armazenar as lições aprendidas, o que compreende o registro de idéias, fatos, questões, pontos de vista, conversas, discussões, decisões etc, que aconteceram no decorrer de um projeto, como mostra a Figura 4. Uma vez que nem todos os relatos são úteis à organização, cabe ao gerente do conhecimento fazer uma triagem dos relatos a serem disseminados e adaptar esse conhecimento, categorizando o problema e a solução adotada.

O artefato é o meio formal, normalmente utilizado para documentar o conhecimento adquirido ao longo do projeto. É importante que esse conhecimento formal seja coletado e compartilhado para que a organização aprenda coletivamente. ProKnowHow permite incluir, alterar e excluir artefatos de projeto, comunicando-se com um sistema de controle de versão.

## **5.3 Manutenção do Processo Padrão**

Uma vez que o processo padrão é a base para a instanciação de processos de projetos, é necessário manter seus ativos (ou componentes). Tomando por base o CMM, o processo padrão do CDSV é composto dos seguintes elementos:

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



**Figura 4 – Cadastro de lições aprendidas**

## 5.4 Disseminação de Experiência

Finalmente, é dado ao desenvolvedor, aquele que executa as tarefas diárias de um projeto e é o responsável pela geração dos artefatos, o acesso à base de experiência. Assim, o conhecimento é disseminado e os desenvolvedores podem receber o apoio que necessitam para executar suas tarefas.

Durante o desenvolvimento, um funcionário pode buscar informações relacionadas a algum procedimento, política, atividade, método, ferramenta ou técnica para apoiar a realização de suas tarefas. Assim, de posse dos comentários e exemplos, ele terá maior embasamento para realizar suas tarefas com sucesso. As figuras 3 e 4 mostram, à esquerda, a área de busca de ProKnowHow.

## 6 Trabalhos Correlatos

Muitas pesquisas relacionadas à melhoria e à instanciação de processos têm sido realizadas pela comunidade de engenharia de software. Várias delas, inclusive, defendem o gerenciamento do conhecimento como fator fundamental para o aumento da qualidade e produtividade. Vale destacar, portanto, que uma característica marcante do trabalho apresentado neste artigo é o fato de adaptar os conceitos desenvolvidos em pesquisas e aplicá-los em um caso real, o CDSV, tendo por base um processo padrão nível 3 do CMM.

Em [2], foi proposta a ferramenta ASSIST-PRO para auxiliar a instanciação de processos de software de acordo com as características específicas de um projeto. Machado [5] propôs um modelo de especialização do processo de software tendo como base a norma ISO 12207 [4] e desenvolveu a ferramenta DEF-PRO para apoiar a instanciação de processos com base nessa norma. Esses dois trabalhos, contudo, não consideram o uso de um processo padrão como base para a definição, mas sim uma ontologia de processo de software no primeiro caso e uma norma de qualidade no segundo. A mesma ontologia usada em [2] foi utilizada no presente trabalho para estruturar a base de experiências. Além disso, não há apoio baseado em experiências anteriores.

No que se refere à abordagem de gerência de conhecimento, Markkula [10] e Althoff et al. [15], assim como o presente trabalho, utilizam essa tecnologia como mecanismo de apoio à disseminação de conhecimento em engenharia de software, armazenando e reutilizando as melhores práticas aprendidas por diferentes projetos. Contudo, o enfoque não é centrado na melhoria de processo, nem há apoio à instanciação de processos de software. No trabalho aqui proposto, uma experiência está sempre associada a um ativo do processo padrão e deve passar por uma adaptação antes de ser disponibilizada em nível organizacional.

## 7 Conclusão

Qualquer tipo de experiência pode ser reutilizado com intuito de evitar retrabalho e melhorar a qualidade, entre os quais, conhecimento formal (artefatos do desenvolvimento) – planos, especificações, código e materiais de treinamento – e conhecimento informal – melhores práticas em engenharia de software e lições aprendidas [13, 15].

Neste contexto, a utilização e a realimentação do processo padrão não são tarefas simples e requerem um suporte baseado na experiência. Para tornar efetivas essas atividades tão importantes para a garantia da qualidade, é necessário oferecer apoio automatizado.

Neste artigo, discutiu-se o compartilhamento de conhecimento e experiência como mecanismo de apoio à melhoria contínua de processos de software e foi apresentada ProKnowHow, uma ferramenta de apoio à melhoria contínua de processos de software, que dispõe de uma base de experiência, contendo ativos do processo padrão da organização, bem com lições aprendidas no uso desse processo. É importante explicitar a importância desses repositórios, pois eles propiciam a disseminação das falhas e sucessos da organização, permitindo o aprendizado coletivo da mesma.

Essa ferramenta está em fase de implantação no CDSV, organização utilizada como base para o desenvolvimento da ferramenta. Em breve os resultados práticos da utilização da ferramenta estarão sendo coletados e avaliados.

## Referências

- [1] Borges, Ligia S. e Falbo, Ricardo A., *Medias Empresas: Atingindo o Nível 3 do CMM*, Anais da X Conferência Internacional de Tecnologia de Software, X CITS, Curitiba - Paraná, Brasil, Maio 1999.
- [2] Falbo, Ricardo A., *Integração de Conhecimento em um Ambiente de Desenvolvimento de Software*. Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, Brasil, 1998.
- [3] Humphrey, Watts S., *Managing the Software Process*. Addison-Wesley Publishing, Company, Massachussets, 1990.
- [4] ISO/IEC 12207, 1995, *Information Technology – Software Life-Cycle Processes*.

- [5] Machado, L. F. D. C., *Modelo para Definição de Processos de Software na Estação TABA*. Tese de Mestrado COPPE/UFRJ, Rio de Janeiro, Brasil, 2000.
- [6] ISO/IEC TR 15504, 1998, *Parts 1-9: Information Technology - Software Process Assessment*.
- [7] Paulk, M.; Weber, C.; Garcia, S.; Chissis, M. e Bush, M., *Key Practices of the Capability Maturity Model*. CMU/SEI-93-TR-25, The Clearinghouse of Computer Science Technical Report, Pennsylvania, 1993.
- [8] O'Leary, D. E., *Enterprise Knowledge Management* IEEE Computer, Março 1998.
- [9] O'Leary, D. E., *Knowledge Management Systems: Converting and Connecting* IEEE Computer, Maio/Junho 1998.
- [10] Markkula, M., *Knowledge Management in Software Engineering Projects*. Proceedings of the 11th International Conference on Software Engineering and Knowledge Engineering, SEKE'99.
- [11] Basili, Victor R. e Caldiera, Gianluigi, H. Dieter Rombach, *The Experience Factory* Volume 1 of Encyclopedia of Software Engineering, Chapter X, John Wiley & Sons.
- [12] Wangenheim, C.G.; Althoff, K.D. e Barcia, R.M., "Intelligent Retrieval of Software Engineering Experienceware". Proceedings of the 11th International Conference on Software Engineering and Knowledge Engineering, SEKE'99, Kaiserslautern, Germany, June 1999.
- [13] Broomé, Mikael e Runeson, Per; *Technical Requirements for the Implementation of an Experience Base* Proceedings of the 11th International Conference on Software Engineering and Knowledge Engineering, SEKE'99.
- [14] Lima, Karina V. C.; Rocha, Ana Regina C. e Travassos, Guilherme H., *Ambientes de Desenvolvimento de Software Orientados à Organização*. COPPE/UFRJ, Rio de Janeiro, Brasil, 2000.
- [15] Althoff, Klaus-Dieter; Birk, Andreas; Hartkopf, Susanne; Muller, Wolfgang; Nick, Markus ; Surmann, Dagmar & Tautz, Carsten, *Managing Software Engineering Experience for Comprehensive Reuse*, Proceedings of the 11th International Conference on Software Engineering and Knowledge Engineering, SEKE'99, Kaiserslautern, Germany, June 1999.