

PARREIRAS, F. S.; BAX, M. P. KMUp: Um Portal para Gestão do Conhecimento no Processo de Desenvolvimento de Software. In: Workshop de Tecnologias da Informação e Gerência do Conhecimento, 3, 2005, Rio Grande do Sul. Disponível em <<http://www.fernando.parreiras.nom.br/publicacoes/KMUp.pdf>>

KMUp: Um Portal para Gestão do Conhecimento no Processo de Desenvolvimento de Software

Fernando Silva Parreiras
Universidade Federal de Minas Gerais
Escola de Ciência da Informação
Programa de Pós-Graduação em Ciência da Informação
Salas 2003/2007
Av. Antônio Carlos, 6627 - Pampulha
Belo Horizonte - MG
CEP 31270-901
fparreiras@eci.ufmg.br

Marcello Peixoto Bax
Universidade Federal de Minas Gerais
Escola de Ciência da Informação
Programa de Pós-Graduação em Ciência da Informação
Salas 2003/2007
Av. Antônio Carlos, 6627 - Pampulha
Belo Horizonte - MG
CEP 31270-901
bax@eci.ufmg.br

Tema do trabalho: Gestão do Conhecimento no Desenvolvimento de Software

KMUp: Um Portal para Gestão do Conhecimento no Processo de Desenvolvimento de Software

Resumo

Diversas ferramentas têm sido utilizadas no processo de desenvolvimento de software com o objetivo de melhorar o uso do conhecimento em um processo tão complicado. Muitas dessas ferramentas se baseiam em trabalhos sobre portais corporativos. Por meio do KMUp, este artigo pretende apontar caminhos na direção do uso integrado de tecnologias como Internet, Intranet, Groupware, Banco de Dados, Ontologias, Agentes, Web Semântica, etc., para a construção de um ambiente extensível e de código aberto. Apresenta-se a arquitetura do KMUp, descrevendo suas principais características.

Abstract

Several tools have been used in software development process, trying to increase the use of knowledge involved in a so complex process. Many of these tools are based in the corporate portals' literature. Presenting KMUp, this paper intends to point directions for a integrated use of technologies like Internet, Intranet, Groupware, Databases, Ontologies, Agents, Semantic Web, etc., in construction of an extensible and open source environment. It presents KMUp framework, by describing its characteristics.

Palavras-chave: gestão de conteúdo; gestão do conhecimento; engenharia de software; Praxis; KMUp.

1. Introdução

A engenharia de software é um domínio extremamente dependente e orientado ao conhecimento. Seus fatores de sucesso estão relacionados com a experiência das pessoas envolvidas nas diversas fases do processo [1]. Na maioria dos casos, as equipes de desenvolvimento não se beneficiam do conhecimento explicitado no processo tanto quanto poderiam e acabam repetindo erros que poderiam ser evitados [2]. Com o objetivo de melhorar o uso do conhecimento, diversas ferramentas têm sido utilizadas, em geral baseadas no conceito de portais. Um portal possibilita a gestão de conteúdo, que traz características como gestão de usuários, permissões, controle de versão, etc. [3].

Além dessas características, a modularidade e a capacidade de integração fazem dos portais ferramentas adaptadas à gestão do conhecimento nos processos organizacionais. O KMUp é um portal voltado à gestão do conteúdo gerado em processos baseados no Processo Unificado [10], projetado para incorporar facilmente novos módulos e integra-se com as outras ferramentas utilizadas ao longo do processo.

Neste artigo, o item 1.1 apresenta a situação problemática e os estímulos para o desenvolvimento deste trabalho. A seção 2 traz o embasamento teórico do trabalho, apresentando o conceito de gestão de conteúdo e o processo Praxis, que ilustra um processo baseado no Processo Unificado, além de analisar os trabalhos relacionados. A seção 3 apresenta o KMUp, descrevendo sua arquitetura e os aspectos relacionados à implementação. Na seção 4 apresentam-se as limitações e as considerações finais acerca desta abordagem.

1.1. Situação problemática

O surgimento de diversas tecnologias causou um grande impacto na gestão do conhecimento nos processos organizacionais. Dentre elas tem-se Internet, *Intranet*, *Groupware* [6], Banco de Dados, Ontologias [7], Agentes Inteligentes, Web Semântica [4], etc.

Contudo, a adoção destas tecnologias de forma não integrada conduziu a implementação de ferramentas divergentes, levando à criação de “ilhas de informação” dentro das organizações. No caso específico da engenharia de software, outras ferramentas também são utilizadas para a produção dos artefatos como compiladores, editores de texto, ferramentas de modelagem, etc.

Neste cenário, emerge a seguinte questão: É possível a implementação de um ambiente, que integre essas tecnologias, sem encapsular ferramentas eficientemente utilizadas?

Por meio do KMUp, este artigo pretende apontar caminhos na direção de soluções para o problema acima.

2. Embasamento teórico-conceitual

Dois conceitos são essenciais ao tema deste trabalho: gestão de conteúdo e o processo Praxis, um processo baseado no Processo Unificado.

2.1. Gestão de conteúdo

A Gestão de conteúdo vem sendo denominada como uma categoria de soluções de software para gestão da informação semi e não estruturada nas organizações.

A definição da *Association for Information and Image Management* (AIIM)¹ para o significado em inglês de Gestão de conteúdo – *Enterprise Content Management* (ECM) – é “as tecnologias, ferramentas e métodos usados para captar, gerenciar, armazenar, preservar e distribuir conteúdo pela empresa”. No nível mais elementar, as ferramentas e estratégias de ECM permitem o gerenciamento de informação estruturada e não-estruturada de uma organização.

Uma solução de gestão de conteúdo oferece acesso ao conteúdo da empresa por meio de uma interface única, baseada em navegador. As funcionalidades essenciais, dentre muitas outras, que caracterizam as soluções e que se desenvolvem à medida que novos produtos chegam à maturidade são [3]:

- a) Gestão de usuários e dos seus direitos (autenticação, autorização, auditoria);
- b) Criação, edição e armazenamento de conteúdo em formatos diversos (html, doc, pdf etc);
- c) Uso intensivo de metadados ou propriedades que descrevem o conteúdo;
- d) Controle da qualidade de informação, com fluxo ou trâmite de documentos;
- e) Classificação, indexação e busca de conteúdo;
- f) Gestão da interface com os usuários (usabilidade, arquitetura da informação);
- g) Sindicalização, publicação da informação em formatos XML visando seu agrupamento ou agregação de diferentes fontes;
- h) Gestão de configuração (controle de versões);
- i) Gravação das ações executadas sobre o conteúdo para efeitos de auditoria e possibilidade de desfazê-las em caso de necessidade.

¹ <http://www.aiim.org/>. A AIIM é uma organização que se dedica a ajudar seus associados a compreender os desafios associados à gestão de documentos, conteúdo e processos de negócio.

Entre as diversas aplicações da gestão de conteúdo estão sítios editoriais e portais corporativos. [8] apresentam casos de uso de portais corporativos em processos de software. Um exemplo de processo de desenvolvimento de software baseado no Processo Unificado é o Praxis, apresentado na próxima seção.

2.2. O Processo Praxis

O Praxis é um processo de desenvolvimento de software, baseado no Processo Unificado, desenhado para dar suporte a projetos didáticos, em disciplinas de engenharia de software de cursos de informática e em programas de capacitação profissional em processos de software. A sigla Praxis significa PROcesso para Aplicativos eXtensíveis Interativos, refletindo uma ênfase no desenvolvimento de aplicativos gráficos interativos, baseados na tecnologia orientada a objetos [9].

O Praxis propõe um ciclo de vida composto por fases que produzem um conjunto precisamente definido de artefatos (documentos e modelos). Para construir cada um desses artefatos, o usuário do processo (estudante ou engenheiro de software) precisa exercer um conjunto de práticas recomendáveis da engenharia de software. Na construção desses artefatos, o usuário do processo é guiado por padrões e auxiliado pelos modelos de documentos e exemplos constantes do material de apoio.

O Praxis se destina a processos de desenvolvimento de software. Para outros tipos de processos de software são necessárias atividades correlatas ao desenvolvimento de software, como manutenção de software, contratação de software, fornecimento de software, desenvolvimento de sistemas e até desenvolvimento de processos.

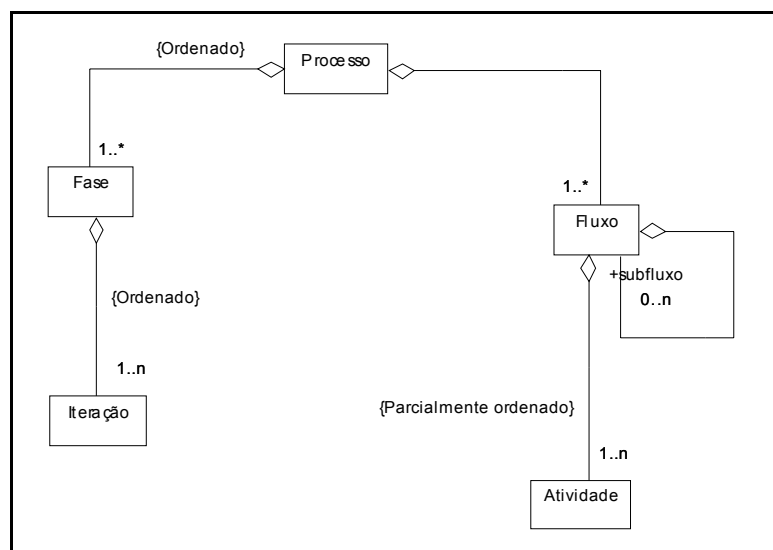


Figura 1. Elementos do Praxis [9].

O Praxis adota as mesmas fases (divisões orientadas para gestão dos projetos) e os mesmos fluxos (divisões orientadas por disciplina da engenharia de software) do Processo Unificado. Uma fase é composta por uma ou mais iterações. Um fluxo é dividido em uma ou mais atividades. Iterações e atividades são exemplos de passos. Essas relações são demonstradas por meio do diagrama de classes UML na Figura 1.

Os principais elementos constituintes do Praxis são:

- Passo: divisão formal de um processo, com pré-requisitos, entradas, critérios de aprovação e resultados definidos
- Fase: divisão maior de um processo, para fins gerenciais, que corresponde aos pontos principais de aceitação por parte do cliente.

- c) Interação: passo constituinte de uma fase, no qual se atinge um conjunto bem-definido de metas parciais de um projeto.
- d) Script: conjunto de instruções que define como uma iteração deve ser executada.
- e) Fluxo: subprocesso caracterizado por um tema técnico ou gerencial.
- f) Subfluxo: Conjunto de atividades mais estreitamente correlatas, que faz parte de um fluxo maior.
- g) Atividade: passo constituinte de um fluxo.
- h) Técnica: método ou prática aplicável à execução de um conjunto de atividades.

Devido ao seu sucesso, o Praxis é utilizado em organizações de diversos tamanhos situadas principalmente no Estado de Minas Gerais. Ferramentas de apoio têm sido desenvolvidas, inclusive para suporte a modelos de qualidade como o CMMI. A próxima seção apresenta alguns trabalhos nesta direção.

2.3. Trabalhos relacionados

O uso de sistemas de gestão de informações no apoio a processos de software tem sido explorado em diversos trabalhos. Esta seção analisa os três mais relevantes para este contexto.

[11] concebeu o Praxis Mentor, que é uma ferramenta CASE que oferece suporte a todas as fases do processo Praxis, instruindo o desenvolvedor no uso do mesmo, fornecendo acesso aos documentos e modelos utilizados em cada fase. Permite controle do fluxo de trabalho, comunicação interna, controle de fases e tarefas do projeto, e acesso ao material de referência do processo. Sua maior restrição é quanto à capacidade de extensão. Todos os módulos foram desenvolvidos de forma proprietária, sem interagir com ferramentas existentes ou utilizar padrões de mercado.

O ODE (*Ontology-based software Development Environment*) é um ambiente de desenvolvimento de software centrado em processo, desenvolvido no Laboratório de Engenharia de Software da Universidade Federal do Espírito Santo (LabES/UFES), tendo por base ontologias [5]. Ele envolve a criação, captura, recuperação, acesso, disseminação, uso e manutenção do conhecimento em torno da memória organizacional. Sua vantagem está no uso de ontologias para o domínio de conhecimento e no uso de agentes. Contudo, o ambiente não fornece interface web.

Já o e-WebProject utiliza a interface web como meio de comunicação [14]. É um ambiente de Desenvolvimento de Software centrado em processos, e apresenta com conceitos fundamentais: o conceito de trabalho cooperativo; a utilização de agentes autônomos; a capacidade de integrar as informações organizacionais e dos projetos; a utilização da Web/Internet como meio de comunicação; e a utilização de abordagens como CMM e SPICE para a referência de processos. Uma de suas principais restrições encontra-se no fato de não se caracterizar como um software de código aberto.

O KMUp se diferencia dos trabalhos acima tendo como principais características: interface web; uso de sistemas abertos e adotados por grande número de desenvolvedores; uso de ontologias para a evolução do domínio de conhecimento; suporte a web semântica.

3. KMUp

O KMUp é um portal para a gestão do conteúdo gerado em processos baseados no Processo Unificado, projetado para incorporar facilmente novos módulos e integra-se com as outras ferramentas utilizadas ao longo do processo de engenharia de software.

3.1. Arcabouço

A implementação proposta é baseada em softwares de código aberto, disponíveis gratuitamente. A partir de uma busca no sourceforge², o repositório de softwares de código aberto com maior número de projetos hospedados no mundo, é possível identificar os sistemas que apóiam toda a implementação do KMUp. O arcabouço utilizado pelo KMUp é formado pelas ferramentas abaixo:

- a) Sistema Operacional: Linux;
- b) Editor UML: Poseidon UML;
- c) Servidor de banco de dados orientado a objeto: Zope;
- d) Servidor de aplicações: Zope;
- e) Servidor Web: Zope;
- f) Plataforma de Gestão de conteúdo: Plone;
- g) Gerador de esquemas (ferramenta utilizada para converter diagramas UML, armazenados no formato XMI em esquemas): Plone Archetypes.

A Tabela 1 apresenta a arquitetura do KMUp, a partir da associação entre as tecnologias mais recentes e as fases do processo de gestão de conhecimento preconizadas. Na interseção entre tecnologia (coluna) e fase no conhecimento (linha) é apresentada a forma de implementação pelo KMUp.

Tabela 1. Arquitetura do KMUp

FASE DO CONHECIMENTO	INTRANET	GROUPWARE	BANCO DE DADOS	ONTOLOGIAS	AGENTES	WEB SEMÂNTICA
Criação e captura de conhecimento	Usa formulários e arquivos em anexo	Usa workflow		Usa diagramas UML		
Recuperação e acesso ao conhecimento	Usa busca textual			Usa taxonomias		Usa repositório de metadados
Disseminação do conhecimento					Usa personalização do portal	
Uso do conhecimento		Usa lista de discussão, chat, fórum, etc				
Manutenção do conhecimento	Usa controle de versão	Usa avaliação e autoria colaborativa			Usa subscrição	
Memória Organizacional			Usa banco de dados Orientado a Objeto	Usa diagramas UML		

Entre as principais características da arquitetura do KMUp estão:

- a) O uso de banco de dados orientado a objeto, que facilita a geração de metadados, criando páginas web preparadas para a web semântica;
- b) O uso de diagramas UML na representação do domínio de conhecimento, isolando a camada de conhecimento da camada de implementação, diminuindo o impacto de alterações no domínio;

² <http://sourceforge.net>

Os tipos de objetos gerenciados pelo KMUp são:

- a) Artefatos: códigos executáveis, códigos fontes, modelos, relatórios, documentos, manuais, propostas, etc.
- b) Lições aprendidas: base de conhecimento e fórum.
- c) Melhores práticas: *FAQ*.
- d) Discussões: *chat*, fórum e listas de discussão.
- e) Bases de especialistas: *blogs* e páginas pessoais.
- f) Repositório de falhas: controle, monitoramento e rastreamento de falhas e correções.
- g) Notícias: eventos, lançamentos, etc.
- h) Mensagens eletrônicas: correspondências trocadas ao longo do projeto.

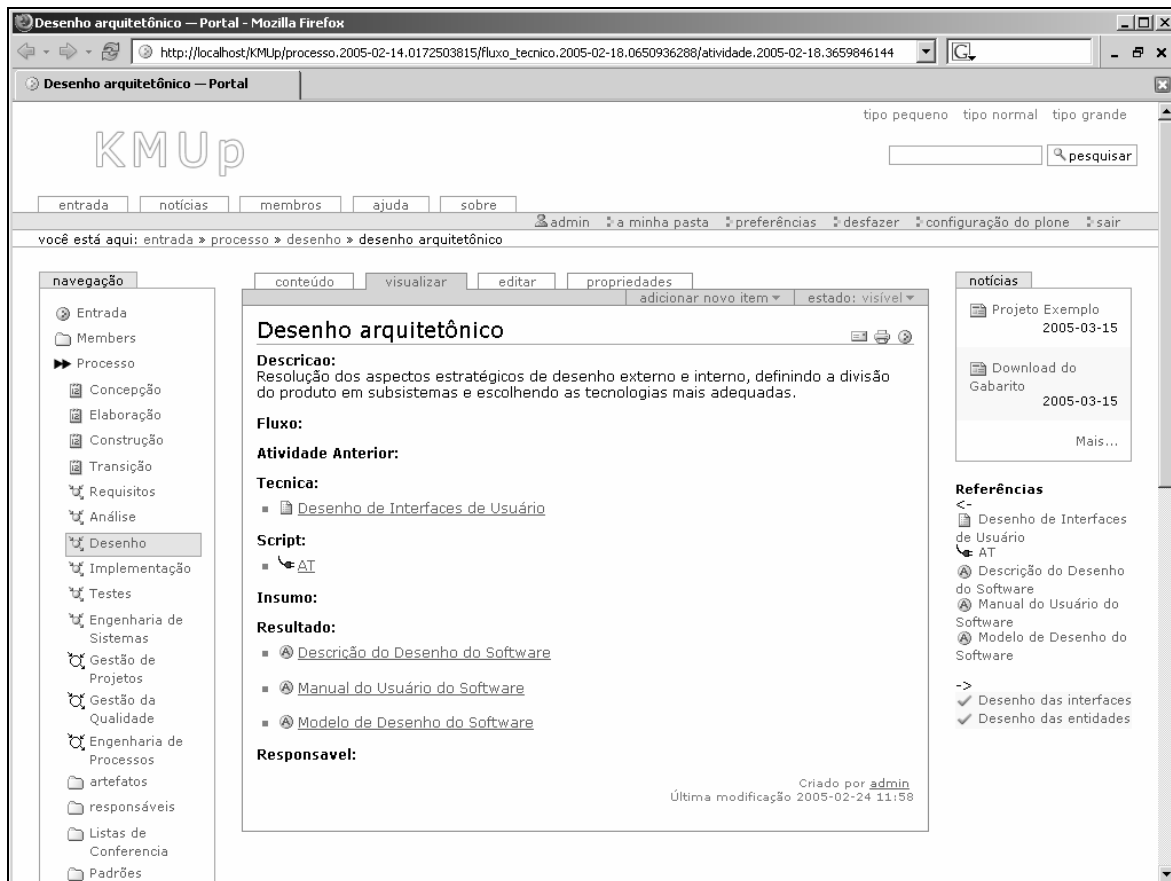


Figura 3. Interface do KMUp

A Figura 3 apresenta a interface protótipo do KMUp. Do lado esquerdo o usuário tem uma visão hierárquica do portal, e do lado direito são apresentadas as referências a um determinado item (fase, atividade, fluxo, etc.). Espera-se que esse tipo de navegação ofereça visões distintas da informação, aumentando as chances de entendimento e recuperação da informação.

4. Considerações Finais

O projeto KMUp está em evolução contínua e todos os interessados podem participar do desenvolvimento acessando o sítio do projeto no repositório brasileiro CódigoLivre³. O desenvolvimento está direcionado, neste momento, para dar suporte aos indicadores do

³ <http://codigolivre.org.br/projects/praxiscmf/>

modelo CMMI níveis 2 e 3. Assim, o KMUp se configura em uma aplicação no auxílio às empresas interessadas nas certificações CMM, CMMI e mpsBr [13].

As ferramentas de código aberto disponíveis ainda não suportam todo o formalismo de modelagem. A impossibilidade de usufruir de todos os recursos da linguagem UML, restringe o potencial de representação e isolamento do domínio de conhecimento.

Um exemplo é a utilização, no experimento, de apenas um dentre os nove diagramas da linguagem UML. Diagramas com o poder de expressão comportamental, como o diagrama de atividades, diagrama de estado e diagrama de sequência, ainda não são suportados, o que acaba restringindo o escopo de aplicação do KMUp em sua versão atual. A representação de fluxos de trabalho também ainda não pode ser representada.

Restringindo-se à utilização de diagramas de classe, outras limitações aparecem como, por exemplo, a incapacidade dos esquemas pré-construídos utilizados de expressar o relacionamento muitos-para-muitos. Logo, quando ocorre este tipo de relacionamento, o código tem de ser alterado manualmente, pelo programador, e caso surja uma classe deste relacionamento, esta não poderá ser representada no diagrama de classe. Por exemplo, no relacionamento Script-Atividade, houve alterações nos códigos fontes para demonstrar que um script possui várias atividades e uma atividade possui vários scripts. Estas questões demonstram que o ambiente ainda precisa evoluir para se tornar uma ferramenta aplicável a projetos mais complexos.

Algumas ferramentas utilizadas no experimento, de código aberto, apresentam restrições quanto ao grau de amadurecimento necessário para aplicações corporativas. Logo, um esforço considerável de desenvolvimento é necessário para a implementação do que seria um produto final.

Referências

1. Desouza, K. C. Barriers to Effective Use of Knowledge Management Systems in Software Engineering. *Communications of the ACM*, vol. 46, n. 1, p. 99-101, jan. 2003.
2. Rus, I., Lindvall, M. Knowledge Management in Software Engineering, *IEEE Software*, May/June 2002, pp. 26 - 38
3. Bax, M. P., Parreiras, F. S. Gestão de conteúdo com softwares livres. In: *KM BRASIL*, 2003, São Paulo. *Anais (CD-ROM)*... São Paulo: [s.n.], 2003.
4. Baader, F. *The description logic handbook: theory, implementation, and applications*. New York: Cambridge University Press, 2002. 555p.
5. Bertollo, G., Ruy, F.B., Mian, P.G., Pezzin, J., Schwambach, M., Natali, A.C.C., Falbo, R.A. ODE – Um Ambiente de Desenvolvimento de Software Baseado em Ontologias. *Anais do XVI Simpósio Brasileiro de Engenharia de Software - Caderno de Ferramentas*, Gramado, Outubro de 2002.
6. Togneri, D. F. Apoio Automatizado à Engenharia de Requisitos Cooperativa. *Dissertação de Mestrado, Mestrado em Informática*, UFES, 2002.
7. Staab, S, Studer, S (eds.). *Handbook on Ontologies*. *International Handbooks on Information Systems*, Springer Verlag, 2004.
8. Parreiras, F. S., Brandão, W. C. O Uso de Portais Departamentais no Processo de Desenvolvimento de Software: O Caso da Unisys Brasil. *Informação & Sociedade: Estudos*, v.14, n.2, 2004.
9. Paula-Filho, W. P. *Engenharia de software: fundamentos, métodos e padrões*. 2. ed. Rio de Janeiro: LTC, 2003. 602p.
10. Jacobson, I., Rumbaugh, J., Booch, G. *Unified Software Development Process*. Reading - MA: Addison-Wesley, 1999.

11. Carvalho, J. G. Praxis Mentor: Uma ferramenta de apoio à utilização de um processo de desenvolvimento de software. Dissertação, Mestrado em Ciência da Computação, UFMG, 2001.
12. Beale, T. Archetypes: Constraint-based domain models for future-proof information systems. In: OOPSLA, 17, 2002, Seattle. Workshop on behavioural semantics, Seattle: ACM. 2002.
13. Weber, K. C. et al. Modelo de Referência para Melhoria de Processo de Software: uma abordagem brasileira. In: Conferencia Latinoamericana de Informatica, 30, 2004, Arequipa.
14. Sant'anna, N. et al. "e-WebProject": Um ambiente integrado para apoio ao desenvolvimento e gestão de projetos de Software. In: XIII Congresso Internacional de Tecnologia de Software, 13, 2002, Curitiba.