

Planejamento e Execução de Gerência do Conhecimento em um Ambiente de Desenvolvimento de Software

Jadielly F. Oliveira, Gabriela F. Andrade, Lorena C. Tavares, Carla A. Lima Reis

Programa de Pós-Graduação em Ciência da Computação (PPGCC)
Universidade Federal do Pará (UFPA)
Rua Augusto Correa 01 – Belém – PA – Brasil

{jadielly,gabriela,clima,lorena}@webapsee.com

Abstract. *Software development processes produce a large amount of knowledge that should be used by organizations to improve software and product quality. This paper presents a software tool support for knowledge management planning and execution, integrated to a Process-Centered Software Engineering Environment.*

Resumo. *Processos de desenvolvimento de software geram grande quantidade de conhecimento que deve ser aproveitado pela organização para melhoria da qualidade do processo e conseqüentemente do produto a ser desenvolvido. Este artigo apresenta um apoio ferramental ao planejamento e execução de estratégias para gerenciar conhecimento, integrada a um ambiente de desenvolvimento de software centrado em processos.*

1. Introdução

Um processo de desenvolvimento de software requer intenso fluxo de conhecimento para a execução das suas atividades. Muitas vezes esse conhecimento é perdido, seja com a saída de membros da equipe, seja com o esquecimento ao passar o tempo. Com isso a organização perde o que vem sendo considerado, atualmente, um dos seus principais capitais que é o intelectual [Rus e Lindvall 2002]. Então, erros que poderiam ser evitados voltam a acontecer, assim como problemas já solucionados voltam a ser repensados por não existirem registros da solução. A organização diminui a qualidade do processo de desenvolvimento e conseqüentemente aumenta o tempo e os custos de seus projetos, entre outros problemas.

Gerenciar o conhecimento “é um processo que ajuda as organizações a encontrar, selecionar, organizar, disseminar e transferir as informações importantes e as experiências necessárias para as várias atividades da organização, tais como, resolução de problemas, aprendizagem dinâmica, planejamento estratégico e tomada de decisões” [Probst, Raub e Romhardt 1999]. Tendo em vista essa perspectiva, modelos de maturidade de software como o MPS.BR - Melhoria do Processo de Software Brasileiro [SOFTEX 2007] e normas como a ISO/IEC 12207 [ISO 2008] já possuem recomendações e práticas que evidenciam a gerência do conhecimento como atividade necessária para que uma organização atinja maior maturidade e capacidade de desenvolvimento de software.

Muitas organizações optam por utilizar Ambientes de Desenvolvimento de Software Centrados em Processos (do inglês *Process-Centered Software Engineering Environment* - PSEE) [Arbaoui et al. 2002], que possibilitam modelagem e execução de processos de software, permitindo apoio à coordenação das atividades planejadas.

O uso de um PSEE, por si só, traz auxílio à gerência de conhecimento, uma vez que centraliza informações importantes acerca do processo e seus envolvidos e, dependendo de suas funcionalidades, pode facilitar a disseminação de conhecimento. Entretanto os ambientes existentes não contemplam todas as funcionalidades necessárias para coordenar as atividades de um processo de forma integrada à gerência de conhecimento da organização.

Este artigo apresenta uma estrutura de apoio ferramental ao planejamento e execução de estratégias de Gerência de Conhecimento (GC) integrada a um PSEE chamado WebAPSEE [Lima e Reis 2007]. A proposta do trabalho considera que a Gerência do Conhecimento deve ser adotada de forma integrada à execução de processos de software em PSEEs. A sinergia entre essas duas ferramentas pode potencializar a disseminação de conhecimento e a adoção do processo de GC, trazendo conseqüências positivas à qualidade do processo e do produto.

A proposta envolve ainda o uso de características Web 2.0 [O'Really 2005], visando a criação coletiva de informação e conhecimento, além da participação rica e ativa dos usuários, proporcionando um ambiente propício, principalmente no contexto atual em que se encontram diversas organizações que estão dispersas geograficamente e se organizam em equipes virtuais, além da questão da terceirização de serviços.

Além desta parte inicial, é apresentada a fundamentação teórica nas seções 2 e 3, o diagnóstico das necessidades na seção 4, a descrição da proposta na seção 5, trabalhos relacionados na seção 6 e por fim as considerações finais na seção 7.

2. Gerência de Conhecimento

Segundo Davenport e Prusak (1999) conhecimento é: “Conjunto de experiências, valores, informações contextuais e percepção de especialistas que proveem uma forma para avaliar e incorporar experiências e informação. É originado e aplicado na mente das pessoas. Nas organizações, geralmente é encontrado não apenas em documentos e repositórios, mas também nas rotinas, processos, práticas e normas organizacionais”.

Existem dois tipos de conhecimento: o conhecimento tácito e o conhecimento explícito [Nonaka e Takeuchi 1995]. O conhecimento tácito é baseado nas experiências das pessoas, é subjetivo, sendo assim difícil de expressar com palavras, sentenças ou números, por exemplo, formalizar as habilidades cognitivas e habilidades técnicas. Já o conhecimento explícito é objetivo e pode ser representado por palavras, sentenças e números. Pode ser representado por documentos, manuais, repositório de dados, entre outros.

Uma gerência de conhecimento eficiente deve ser capaz de apoiar a criação, captura e utilização dos vários tipos de conhecimento. Na engenharia de software este conhecimento pode ser de diversos tipos: **técnico, gerencial, de domínio, corporativo, de produto, e do projeto** [Rus e Lindvall 2002].

Existe um conjunto básico de atividades que possibilitam a gerência de conhecimento de forma sistemática, são elas [Rus e Lindvall 2002; Natali 2007]: a) **identificação** que é reconhecer o conhecimento importante; b) **aquisição** que é coletar e armazenar o conhecimento útil à organização; c) **construção** que é gerar um novo conhecimento útil a partir do existente; d) **manutenção** que é remover ou atualizar o conhecimento defasado; e) **disseminação** que é distribuir o conhecimento apropriado

com o cuidado de não sobrecarregar os membros da organização com excesso de informações; f) **utilização** que é aproveitar o conhecimento para aprimorar o aprendizado assim como melhorar a realização das atividades; e g) **valorização** que é ponderar o conhecimento através da medição da sua utilidade e das suas contribuições para a organização.

No modelo MPS.BR os resultados esperados diretamente relacionados à GC são:

- GRH8. Uma estratégia apropriada de gerência de conhecimento é planejada, estabelecida e mantida para compartilhar informações na organização.
- GRH9. Uma rede de especialistas na organização é estabelecida e um mecanismo de apoio à troca de informações entre os especialistas e os projetos é implementado.
- GRH10. O conhecimento é prontamente disponibilizado e compartilhado na organização.

3. Automação de Processos de Software

Ambientes para automação de processos de software (*PSEEs*) caracterizam-se por apoiar a modelagem e execução de processos de modo a auxiliar e controlar todas as atividades envolvidas na produção e manutenção de um software.

Segundo Dowson (1994), existem três diferentes domínios de processos de software: o domínio das definições, o da realização e o da execução de processos. O domínio das definições do processo contém modelos de processos expressos em alguma notação, denominadas Linguagens de Modelagem de Processos (*Process Modeling Language – PML*). O domínio da realização do processo engloba as atividades ou ações conduzidas por pessoas e agentes não humanos (software). O domínio da execução do processo definido preocupa-se com o que é necessário em um ambiente de desenvolvimento de software para suportar a execução de uma definição de processo.

Os domínios da realização e da execução devem estar em conformidade para que os dados do processo sejam armazenados de forma correta no PSEE. Para manter essa conformidade é necessário que de alguma forma seja obtido o *feedback* acerca do estado das atividades realizadas pelos desenvolvedores durante a execução do processo. Em contrapartida o PSEE deve fornecer assistência e suporte à realização do processo.

4. Diagnóstico das Necessidades para Implementação de GC

Com o objetivo de nortear o desenvolvimento da abordagem proposta, além do estudo acerca de métodos, técnicas e ferramentas de apoio à GC, foi realizada uma pesquisa a partir da realização de entrevistas em quatro organizações de desenvolvimento de software regionais, a fim de investigar quais os principais problemas enfrentados em relação à GC, bem como descobrir os requisitos necessários para implementação de estratégias de GC nessas organizações.

A pesquisa foi conduzida com a aplicação de entrevistas semi-estruturadas abertas, com Gerentes, Desenvolvedores Novatos e Desenvolvedores Experientes. O objetivo de utilizar essa divisão de perfis é observar as dificuldades que cada um enfrenta em relação ao fluxo de conhecimento na organização, bem como identificar as

principais necessidades de cada perfil. A subseção seguinte apresenta uma breve descrição das características das organizações nas quais foram aplicadas as entrevistas.

4.1. Caracterização das Organizações

A primeira organização (A) onde foram aplicadas entrevistas é caracterizada principalmente pela baixa rotatividade de pessoas, por ser formada apenas por funcionários concursados. O processo de software utilizado está bem estabelecido, e todos na empresa conhecem e seguem os padrões de ferramentas e tecnologias utilizadas. Vários projetos pequenos são desenvolvidos ao mesmo tempo, normalmente por uma ou duas pessoas. Apenas um gerente controla todos os projetos, e a cada dois meses são realizadas reuniões, nas quais são apresentadas novas tecnologias, ou mudanças que possam ter ocorrido nos processos.

A segunda organização (B) é caracterizada pela alta rotatividade de pessoas, por se tratar de uma instituição acadêmica, formada por alunos de graduação, mestrado e doutorado. Atualmente é constituída por várias equipes que possuem um gerente para cada projeto, no entanto alguns membros trabalham em mais de um projeto. A gerência de processos é realizada com o auxílio de um PSEE, e as equipes utilizam wikis e listas de discussão como ferramentas para compartilhamento de conhecimento. Apenas um dos projetos possui um plano de GC definido, e um processo-padrão com atividades de GC incorporadas.

A terceira organização (C) possui nível F de maturidade do MPS.BR e é formada em sua maioria por várias equipes pequenas que desenvolvem projetos de curta duração, tem um coordenador por equipe, e gerentes que controlam vários projetos ao mesmo tempo. Além desses projetos, existem processos de manutenção de sistemas legados. A organização possui processos-padrão bem definidos para cada tipo de projeto desenvolvido. Atualmente a rotatividade de pessoas é freqüente e espera-se que uma grande quantidade de novos membros seja contratada. Membros novatos estão entrando com freqüência na empresa, que utiliza wikis e e-mail para troca de informações, no entanto já existe um projeto que prevê uma estratégia com cadastro de Lições Aprendidas, mas ainda não é utilizado por todos.

A quarta organização (D) é formada por um número muito grande de funcionários de desenvolvimento e implantação de software. Atuam em dois grandes sistemas instalados em vários pólos distribuídos, com vários projetos de manutenção. Não é comum entrada de novos membros na empresa, no entanto a rotatividade de membros em projetos é freqüente. Utilizam apenas e-mail para troca de conhecimento.

4.2. Análise do Diagnóstico das empresas

Após avaliação das entrevistas realizadas, foi possível verificar uma série de problemas em comum entre as organizações com relação à GC. O principal problema reportado foi a falta de tempo dos desenvolvedores em registrar conhecimento adquirido durante o desenvolvimento de suas atividades. Isso é decorrente do fato de que a gerência não prevê no planejamento do projeto o tempo necessário para que possam ser registrados itens de conhecimento. Outro problema de grande relevância é a falta de ferramentas adequadas para realização dessas atividades, tendo em vista que os desenvolvedores utilizam uma grande quantidade de ferramentas, torna-se inviável inserir mais uma para registro de conhecimento.

Outro problema crítico reportado pelos gerentes foi a dificuldade de gerenciar e distribuir o conhecimento gerado pela equipe. Por exemplo, no caso da organização **A** onde existe apenas um gerente para todos os projetos, haveria uma sobrecarga de trabalho muito grande para essas tarefas. Por isso a organização considerou adequada a estratégia de realizar apresentações a cada dois meses acerca de conhecimento importante para a organização.

Um problema em particular da organização **B** é a diferença dos níveis de experiência entre os membros dos projetos, que por se tratar de uma organização acadêmica, possui alocação de membros baseada na área de interesse do aluno, bem como pode ocorrer alocação de pessoas sem experiência por falta de profissionais qualificados (normalmente alunos do início dos cursos). Essas pessoas novatas poderiam ser beneficiadas com conhecimento de outros alunos, caso eles tivessem registrado esse conhecimento, o que minimizaria o desnível de experiência entre os alunos.

As empresas **C** e **D** apresentaram um grande problema em comum em relação à comunicação entre os membros, bem como à dificuldade de saber quem faz o quê dentro da organização, visto que ambas possuem um número grande de pessoas. Muitas vezes ocorre que algumas pessoas ficam sobrecarregadas em tirar dúvidas de outros membros da organização e são interrompidas constantemente, atrasando suas tarefas.

Os principais requisitos identificados a partir da análise das entrevistas foram: a integração de ferramentas de GC ao ambiente de trabalho dos membros da organização para facilitar tanto a gerência como a coleta de conhecimento; o estabelecimento de estratégias de GC adequadas à diversidade de organizações; a aplicação de abordagens para facilitar a busca de pessoas na organização e a comunicação entre elas; o planejamento adequado de tempo para que os desenvolvedores possam registrar conhecimento, e por fim o desenvolvimento de ferramentas para facilitar a distribuição de conhecimento de forma ampla na organização. É importante observar que os requisitos elicitados com a análise das entrevistas foram compatíveis com os resultados esperados do MPS.BR apresentados na seção 2.

5. WebAPSEE Knowledge Manager (WKM)

A ferramenta *WebAPSEE Knowledge Manager* (WKM) é uma extensão ao ambiente WebAPSEE [Lima e Reis 2007] que foi desenvolvida para apoiar o planejamento e a execução de estratégias de GC em organizações de software. O WebAPSEE é um PSEE e por isso permite a integração de vários serviços de gerência de processos, incluindo modelagem, execução, visualização, instanciação e resposta a eventos da execução. O ambiente é composto por um módulo servidor e dois módulos cliente: O *Manager Console*, usado pelos gerentes e a *TaskAgenda*, utilizada pelos desenvolvedores e quaisquer envolvidos no processo.

Através do *Manager Console* (MC) o gerente de projeto pode modelar e reutilizar processos, gerenciar execução de processos, visualizar relatórios de monitoramento e gerenciar informações da organização, como artefatos, pessoas e recursos, através de um editor gráfico. Na *TaskAgenda* os desenvolvedores visualizam suas tarefas juntamente com os artefatos de entrada e de saída e podem informar o estado de suas atividades, ou quando possível, delegar atividades a outros membros do projeto. As informações registradas na *TaskAgenda* são enviadas automaticamente para

o *Manager Console* possibilitando que os gerentes possam acompanhar o andamento do processo em tempo real.

Atualmente, encontra-se em desenvolvimento a versão WebAPSEE 2.0, baseada nas características da Web 2.0 que tem como desafio manter a infra-estrutura da versão anterior, reutilizando o código existente (aproximadamente 150 kLOC Java e 200 serviços remotos). O módulo *TaskAgenda* 2.0 foi finalizado incluindo a integração das funcionalidades relacionadas à GC, enquanto que o *Manager Console* 2.0 ainda está em fase de desenvolvimento, por isso o módulo de GC foi implantado na versão 1.4 (*Desktop*) do *Manager Console*.

5.1. Visão Geral da Arquitetura

A abordagem adotada para o desenvolvimento do Módulo WKM foi adaptada a partir do modelo de Hansen (1999), que classifica a estratégia de gestão do conhecimento em dois tipos: codificação e personalização. A estratégia de Codificação focaliza o uso de tecnologia de informação no armazenamento de conhecimento relevante que possa ser facilmente acessado e utilizado por qualquer outra pessoa da companhia. Por outro lado, na chamada estratégia de Personalização, o conhecimento é compartilhado principalmente por meio de contatos diretos de pessoa a pessoa. Neste caso, o propósito da TI não é armazenar conhecimento, mas auxiliar as pessoas a comunicá-lo.

O modelo de Hansen propõe que a organização opte por uma das duas estratégias, no entanto estudos [Desouza et al. 2006] apontam benefícios da adoção de uma abordagem híbrida. Portanto a arquitetura proposta foi elaborada a partir da combinação entre as estratégias de codificação e personalização. Vale ressaltar como ponto forte que a utilização de características da Web 2.0 na *TaskAgenda* facilitou bastante o desenvolvimento de funcionalidades para apoiar e melhorar a comunicação entre os membros dos projetos, como o *chat*, a visualização do organograma e das informações sobre departamentos e a busca de especialistas.

O WebAPSEE possui um meta-modelo de dados que facilita a representação do conhecimento, visto que os principais componentes da arquitetura WebAPSEE são tipados. As hierarquias de tipos relacionadas aos componentes (atividades, artefatos, recursos, cargos) permitem que sejam descritos processos abstratos que podem ser refinados conforme necessário para execução, ou usados para reutilização, e ainda apóiam a descrição de mecanismos que lidam com elementos genéricos de processos.

Seguindo o padrão descrito acima, o componente “conhecimento” foi integrado à hierarquia de tipos do WebAPSEE representado pela classe *KnowledgeType* no modelo de dados ilustrado na Figura 2. A definição dos subtipos foi baseada na classificação apresentada em [Rus e Lindvall 2002]: **técnico, gerencial, de domínio, corporativo, de produto, e do projeto.**

Na arquitetura proposta o componente Conhecimento, representado pela classe *KnowledgeItem* pode estar relacionado a outros componentes do WebAPSEE como: atividades, artefatos, ferramentas, recursos, cargos, processos, organização, dentre outros. Dessa forma um item de conhecimento pode ser registrado tanto em nível de projeto como em nível organizacional. Esses componentes são representados no modelo de dados por uma classe genérica denominada *WebAPSEEObject* (Figura 1), que pode possuir vários itens de conhecimento, no entanto um item de conhecimento somente pode estar relacionado a um componente do WebAPSEE. Além disso, um item de

homologação (nesse estado o item não pode mais ser editado). A partir desse estado, o item de conhecimento pode ser enviado para avaliação, ficando no estado “aguardando avaliação”. Após avaliação o item pode ser homologado diretamente passando para o estado “disponível”, pode ser desabilitado ficando no estado “indisponível” ou pode ser enviado para revisão ficando no estado “aguardando revisão”, caso seja aceito para revisão o item retorna ao estado “salvo”.

5.2. Visão Geral das Funcionalidades

A integração de funcionalidades do módulo WKM foi desenvolvida na versão *Desktop* do *Manager Console* e na versão web 2.0 da *TaskAgenda*. O Manager e Agenda interagem entre si através de um servidor que gerencia a execução do processo, e o fluxo de conhecimento na organização. Os desenvolvedores também podem se comunicar através de suas agendas. Todos os itens de conhecimento registrados são armazenados em uma base e os anexos e artefatos relacionados estão em um repositório de dados. A figura 2 ilustra uma visão geral do funcionamento do sistema.



Figura 2. Visão geral do funcionamento do sistema.

5.2.1. Funcionalidades para os Gerentes

O *Manager Console* permite que os Gerentes de Conhecimento da Organização (cargo previamente definido no modelo do WebAPSEE) possam inserir tipos¹ de conhecimento a serem coletados na organização, como Lições Aprendidas, Novas Tecnologias e Melhores Práticas, por exemplo. A estrutura de campos desses tipos pode ser definida dinamicamente, semelhante à abordagem definida por Montoni (2003). A figura 3 lustra a interface correspondente.

Um conjunto de tipos de conhecimento é proposto na ferramenta, no entanto é possível modificá-los conforme necessidade da organização. Esses tipos de conhecimento poderão ser posteriormente vinculados a um ou mais Planos de GC no momento de sua definição junto ao processo de software.

Questões de avaliação devem ser vinculadas aos tipos de conhecimento para que o Gerente possa solicitar a especialistas que emitam pareceres acerca dos itens de conhecimento cadastrados. Essas questões podem ser formuladas independentemente

¹ Não confundir com os meta-tipos predefinidos de conhecimento propostos no modelo do WebAPSEE.

dos tipos para que sejam reutilizadas e podem ser de três formatos: subjetiva, de múltipla escolha ou escolha exclusiva.

Id	Nome	Rótulo	Formato
cat_la	categoria	Categoria da Lição	Caixa de Seleção
titulo_la	titulo	Título	Texto
desc_la	descricao	Descrição	Texto
cont_la	contexto	Contexto do Probl...	Texto
sol_la	solucao	Solução Adotada	Texto
res_la	resultado	Resultado Obtido	Texto

Figura 3. Parte do Modelo de Dados do WKM que representa o conhecimento.

Após a definição dos tipos de conhecimento e das questões de avaliação, os gerentes de projeto podem utilizar essas informações na elaboração de Planos de GC que devem ser vinculados a Projetos no *Manager Console*. Esse plano é composto a partir de cinco etapas:

- **Descrição Textual:** descrição do escopo do plano, justificativa para adoção da estratégia escolhida, bem como outras informações consideradas pertinentes;
- **Seleção de Tipos de Conhecimento:** seleção dos tipos de conhecimento que deverão ser coletados durante o processo. Quando um tipo é vinculado ao Plano de GC, automaticamente é disponibilizado um formulário de cadastro, e outro de avaliação com as questões correspondentes, na *TaskAgenda*. A seleção desses tipos não impede que outros tipos sejam cadastrados durante o projeto, a elaboração do plano serve apenas como um guia;
- **Definição de Marcos para Aquisição de Conhecimento (Figura 4):** o gerente pode definir marcos para aquisição de conhecimento relacionando as atividades com os tipos de conhecimento que devem ser coletados no momento da finalização ou de início da atividade. Na figura, o lado esquerdo mostra a árvore de atividades de um processo de software, onde D representa uma atividade decomposta (em novo sub-processo) e N uma atividade normal (folha).

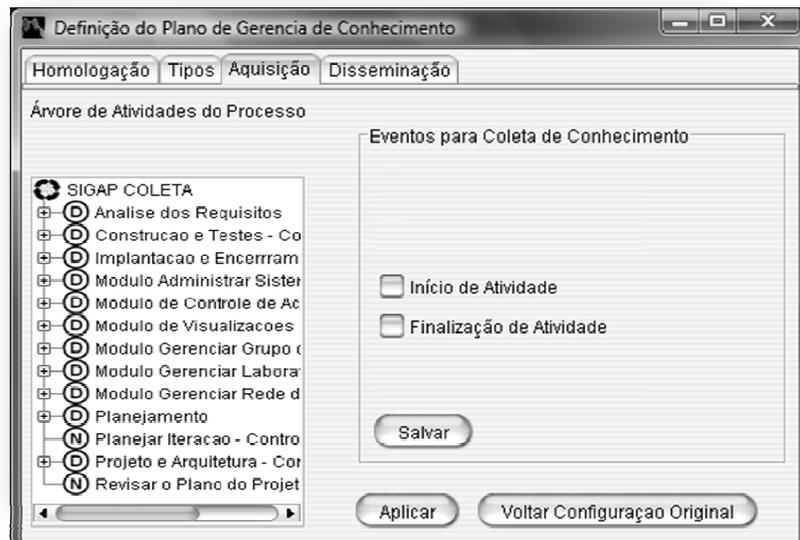


Figura 4. Ferramenta para Definição do Plano de GC (Aba de definição dos marcos para aquisição de conhecimento).

- **Definição de Responsáveis pela Homologação:** a definição dos responsáveis pela homologação deve ser feita de acordo com os tipos, sendo que um tipo só pode ser homologado por uma pessoa, porém uma pessoa pode ser responsável pela homologação de vários tipos. Apenas pessoas com o cargo de Gerente do Conhecimento podem ser responsáveis pela homologação. Essa pessoa pode ou não ser o Gerente do Projeto.
- **Seleção de tipos de conhecimento para disseminação:** o gerente pode selecionar tipos de conhecimento para serem disseminados de forma ativa durante a execução dos processos. Os tipos selecionados para disseminação não precisam ser os mesmos definidos para coleta.

A partir do *Manager Console* é possível também manter os itens de conhecimento disponíveis na base de dados, para isso os gerentes de conhecimento da organização podem definir critérios para manutenção automática de itens de conhecimento, ou podem visualizar as características de cada item e desabilitá-los de acordo com seus próprios critérios. As opções disponíveis para programar a manutenção da base de conhecimento são: O tempo máximo que um item de conhecimento pode permanecer disponível na base de conhecimento sem que seja utilizado por nenhum usuário e o percentual de avaliações negativas feitas pelos usuários do conhecimento.

Por fim, vale ressaltar que o sistema permite a geração de um relatório com todas as informações contidas no Plano de GC, bem como relatórios de desempenho em relação ao número de itens de conhecimento cadastrados por projeto, por agente ou por cargo. A partir desses relatórios é possível à alta diretoria planejar um sistema de recompensas para os agentes que mais cadastram conhecimento, servindo como uma forma de motivação para que mais conhecimento útil para a organização seja coletado.

5.2.2. Funcionalidades da *TaskAgenda* - Web 2.0

Na *TaskAgenda* são concentradas a maioria das funcionalidades e as mais importantes do Módulo WKM, visto que é a partir dessa interface que são inseridos, avaliados e homologados os itens de conhecimento, bem como é na *TaskAgenda* que os usuários aprendem e compartilham informações com outros membros da organização. A figura 5 ilustra um exemplo de utilização do formulário de cadastro de lições aprendidas.

The image shows a screenshot of a web application interface. On the left, there is a sidebar titled 'admin's Task Agenda' with a 'Processes' dropdown set to 'WebAPSEE Web 2' and a 'Tasks' list. The 'Tasks' list includes 'Definir Escopo do Projeto', 'Elaborar Termo de Abertura', 'Planejar Testes' (highlighted), 'Elaborar Modelo de Análise e', 'Detalhar os R', and 'Atualizar a Matriz de Rast'. Below the tasks are fields for 'Delegated From', 'Delegated To', and 'Working Hours' (set to 0). At the bottom of the sidebar are buttons for 'Start', 'Finish', and 'Cancel'. The main area is a form titled 'Cadastro de lições aprendidas'. It contains several sections: 'Meta-Tipo' (dropdown: Técnico), 'Categoria da Lição' (dropdown: Oportunidade de Melhoria), 'Título' (text input: Falta de planejamento para teste de carga), 'Descrição' (text area: Durante o planejamento não foram incluídos os testes de carga mesmo tendo como requisito funcional de alta prioridade o desempenho da ferramenta), 'Contexto do problema:' (text area: Durante a utilização a ferramenta apresentou baixo desempenho quando havia grande quantidade de clientes acessando o servidor. O problema não foi tratado), 'Solução Adotada:' (text area: Planejamento de testes de carga para avaliar o número máximo de conexões suportadas pelo servidor), and 'Resultado Obtido:' (text area: Após a realização dos testes foi possível observar o desempenho do sistema em relação ao número de acessos). At the bottom of the form are buttons for 'Salvar', 'Limpar', 'Voltar', and 'Enviar Para Avaliação'.

Figura 5. Formulário de Cadastro de Lições Aprendidas.

Todos os itens de conhecimento inseridos na *TaskAgenda* são enviados para o gerente responsável pela homologação do tipo de conhecimento (definido no Plano de GC). A homologação de itens também é feita pela *TaskAgenda* (o módulo apenas fica disponível para os responsáveis). Quando necessário, o gerente pode enviar um item de conhecimento para avaliação de especialistas. Para isso o responsável tem o apoio da ferramenta que disponibiliza opções para filtrar o agente mais adequado para avaliação do item. Esse filtro possui critérios como o cargo, as habilidades do agente, o grau das habilidades (definido manualmente pelo gerente no *Manager Console*) e o número de avaliações feitas por cada agente. Depois de homologados, os itens de conhecimento ficam disponíveis na base de dados, podendo ser acessados por todos os membros da organização.

A busca de itens de conhecimento do WKM pode ser realizada a partir de vários filtros: por meta-tipo do conhecimento, por tipo de estrutura (lição aprendida, melhores práticas), por componentes do WebAPSEE (atividades, artefatos), por projetos ou ainda pela inserção de palavras-chave. A utilização da plataforma web 2.0 facilita esse processo de aprendizado, pois proporciona maior flexibilidade na navegação e visualização dos itens de conhecimento.

O mecanismo de busca foi desenvolvido com base no trabalho de Sales (2006), que apresenta uma ferramenta para buscar processos a partir de configurações e combinação de informações, denominada *SearchEngine*. A ferramenta utiliza a técnica de Raciocínio Baseado em Casos para realizar os cálculos de similaridade entre uma consulta fornecida pelo usuário e cada processo existente no repositório de processos do WebAPSEE, extraindo um coeficiente que representa o grau de similaridade entre eles.

Além da busca de conhecimento a ferramenta proporciona a disseminação ativa de itens de conhecimento, ou seja, sem a solicitação do usuário. Atualmente apenas os itens de conhecimento relacionados a atividades são disseminados. Sempre que uma atividade é iniciada na agenda, o sistema verifica a partir de cálculos de similaridade entre os atributos das atividades (tipo, identificador, nome, script e artefatos de entrada e de saída), se existem itens de conhecimento para que possam ser sugeridos como auxílio ao usuário da agenda na realização da tarefa. Caso sejam encontradas atividades similares com mais de um item de conhecimento, o sistema utiliza como critério de ordenação a data de cadastro e as avaliações de utilidade.

A versão 2.0 da *TaskAgenda* possui um chat com possibilidade de registro de conversas e um mini fórum onde são inseridos questionamentos, novidades e informações em geral separados por assunto. Além desse mecanismo de comunicação o sistema permite a busca por pessoas a partir do sistema de páginas amarelas do WebAPSEE, no qual os usuários podem configurar um filtro com as características da pessoa que deseja encontrar, como o cargo, habilidade, ou ainda visualizar o organograma da organização e visualizar as informações de cada departamento e seus integrantes. Também é possível visualizar graficamente na *TaskAgenda* os processos exatamente como modelados no *Manager Console*, possibilitando ao usuário conhecer o processo, as pessoas e os recursos envolvidos, assim como a sequência e dependências entre atividades.

5.3. Análise do Atendimento aos Resultados Esperados do MPS.BR

Durante o desenvolvimento da ferramenta primou-se pelo atendimento do modelo MPS.BR com o objetivo de proporcionar apoio ferramental para que as organizações pudessem alcançar mais níveis de maturidade do modelo. A tabela a seguir apresenta essa análise.

Tabela 1. Atendimento aos resultados esperados do MPS.BR

MPS.BR	Atendimento
GRH8	A ferramenta apóia o atendimento do resultado esperado, pois permite a elaboração de Planos de GC, bem como a definição da estratégia adequada à organização através da definição dinâmica de estruturas para coleta de conhecimento e questões de avaliação. Além disso, o sistema fornece apoio à execução do processo de GC contemplando a aquisição, manutenção, disseminação, utilização e valoração de conhecimento.
GRH9	A ferramenta atende parcialmente o resultado, pois utiliza ferramentas para apoiar a comunicação entre os membros da organização, no entanto, apesar da rede de especialistas estar disponível no Manager Console para o gerente, não está para os desenvolvedores ainda. Portanto, o mecanismo de apoio à formação da rede de especialistas precisa de melhorias no sentido de aumentar o nível de automação e visualização da rede.
GRH10	A ferramenta atende o resultado esperado, pois armazena o conhecimento e disponibiliza funcionalidades de busca e disseminação utilizando características da web 2.0, o que proporciona maior facilidade na navegação e visualização de conhecimento organizacional

6. Trabalhos Relacionados

Várias abordagens para GC vêm sendo apresentadas pela literatura, com diferentes propósitos e domínios de aplicação. Entretanto, apenas quatro delas são comentadas nesse artigo devido a restrições de espaço, maior relação com a proposta apresentada e evidências de real utilização das ferramentas.

O trabalho de Natali (2003) apresenta uma infra-estrutura para GC integrada ao Ambiente de Desenvolvimento de Software ODE (*Ontology-based software Development Environments*), que apóia compartilhamento e reuso de conhecimento através de criação e captura, recuperação e acesso, disseminação, uso e manutenção do conhecimento. No entanto, o escopo do trabalho está restrito apenas a três itens de conhecimento (lições aprendidas, artefatos e instâncias de ontologias) e não aborda aspectos colaborativos da GC, o que é resolvido com o trabalho apresentado por Falbo, Arantes e Natali (2004).

A ferramenta *ProKnowHow* foi desenvolvida com o objetivo principal de apoiar a disseminação do conhecimento sobre processos. São objetivos adicionais: apoiar a instanciação do processo para projetos específicos, divulgar o conhecimento adquirido durante os projetos e apoiar a melhoria contínua do processo de software através da realimentação [Borges e Falbo 2001]. *ProKnowHow* oferece recursos para coleta de experiências e a disseminação destas experiências. A manutenção do conhecimento é feita pelo usuário gerente de processo de forma não automatizada, ou seja, ele tem que verificar todo conhecimento inserido e avaliar o quanto aquele conhecimento é relevante para a organização. A ferramenta também não possui uma forma ativa de disseminação do conhecimento.

O ambiente CORE-KM (*Customizable Organizational Resources Environment with Knowledge Management*) é um ambiente customizável para gerência de conhecimento em diferentes organizações, capaz de apoiar seus processos organizacionais [Galotta, Oliveira e Rocha 2004]. Integrada ao Ambiente de Desenvolvimento de Software Estação TABA, a ferramenta CORE-KM visa apoiar a aquisição, uso, construção, disseminação, valoração e manutenção do conhecimento, além de manter uma memória organizacional. A aquisição do conhecimento é feita por meio de uma ferramenta chamada ACKNOWLEDGE [Montoni 2003]. A disseminação é feita de forma passiva, ou seja, o usuário busca pelo conhecimento, faltando então uma forma de divulgação que forneça conhecimento relevante de acordo com o perfil do usuário, no momento adequado e no formato ideal.

O principal diferencial da proposta apresentada nesse artigo está no alto nível de automação e flexibilidade, visto que o ambiente possibilita a criação dinâmica de estruturas de conhecimento e questões de avaliação, permite a elaboração de planos de GC de acordo com as necessidades de cada projeto, realiza disseminação ativa de itens de conhecimento relacionados à atividades e permite a definição de critérios para a manutenção automática da base a partir da valoração de itens de conhecimento. Além disso, fornece apoio a todas as atividades desejáveis em um processo de GC, citados na seção 2, que são: **identificação, aquisição, construção, manutenção, disseminação, utilização e valoração de conhecimento.**

Outro ponto forte do trabalho é a utilização de hierarquias de tipos relacionadas aos componentes do WebAPSEE, que permitem maior refinamento na classificação de

conhecimento, facilitando a busca, a disseminação, bem como possibilitando que conhecimentos não relacionados a um processo de software específico sejam tratados de forma genérica sem perder o benefício de possuir uma classificação adequada. É importante frisar também que a ferramenta desenvolvida foi integrada a um ambiente que permite a execução do processo e o acompanhamento das atividades e mudanças em tempo real.

Por fim, as ferramentas e elementos gerados na execução do projeto aqui proposto estão contextualizados na *Web 2.0*, diferenciando-as de qualquer ferramenta citada anteriormente, e por isso tornando inviável a integração de outras ferramentas existentes na academia e no mercado ao ambiente WebAPSEE. Além disso, a ferramenta proposta possui a integração de várias funcionalidades, relacionadas tanto a estratégia de codificação como de personalização, o que traria a necessidade de integração de diferentes ferramentas e modelos tornando o processo de desenvolvimento complexo e propenso a muitos erros.

7. Conclusões e Trabalhos Futuros

Este artigo apresentou a definição e desenvolvimento de uma estrutura de apoio ferramental ao planejamento e execução de estratégias de GC em processos de software, integrada a um PSEE denominado WebAPSEE. Também foi fornecido o referencial teórico para o embasamento da proposta.

A principal contribuição deste trabalho consiste em proporcionar flexibilidade e automação em alto nível, além de integrar várias funcionalidades em um único ambiente de desenvolvimento de software que permite execução do processo. Além disso, a ferramenta atende à maioria dos resultados esperados do MPS.BR, favorecendo organizações que desejam obter o nível E de maturidade.

Como oportunidade de melhoria destaca-se o desenvolvimento de algoritmos mais eficientes para busca e disseminação de conhecimento, bem como para a disponibilização automática de informações sobre especialistas da organização. Em especial, serão exploradas características como redes sociais e colaboração para disseminação e compartilhamento do conhecimento, *folksonomia* para categorização e busca, e *mashup* para possibilitar que conteúdos e ferramentas possam ser combinados em uma única interface.

Com o objetivo de analisar a utilização da ferramenta WKM, está em andamento um estudo de caso conduzido em uma organização Nível G do MPS.BR que já utiliza a versão 1.4 (Desktop) do ambiente WebAPSEE. O estudo será conduzido a partir da realização de cinco etapas: diagnóstico da organização, elaboração da estratégia para GC, implantação da nova versão da ferramenta (com o módulo WKM integrado, preservando os dados armazenados na base dos processos já executados com versão anterior da ferramenta), observação e acompanhamento na execução dos processos e por fim análise dos resultados. Como trabalho futuro pretende-se implantar a ferramenta em mais organizações para avaliar sob diferentes aspectos e em diferentes contextos, quão útil a GC pode ser no desenvolvimento de processos de software.

8. Agradecimentos

Os autores do artigo agradecem à FAPESPA e ao CNPQ pelo apoio financeiro que possibilitou a realização desse trabalho.

9. Referências

- Arbaoui, S.; Derniame, J.; et al.;(2002)“A comparative review of Process-Centered Software Engineering Environments”. *Annals of Software Engineering*, p. 311-340.
- Borges, L. M. S.; Falbo, R. A. (2001), “Gerência de Conhecimento Sobre Processo de Software”. *Simpósio Brasileiro de Engenharia de Software*, pp. 27-38, RJ, Brasil.
- Davenport, T., Prusak, L. (1999), “Conhecimento Empresarial”. RJ: Ed.Campus.
- Desouza, K. C., Awazu Y., and Baloh. P. (2006) “Managing Knowledge in Global Software Development”, *Efforts: Issues and Practices. IEEE Software*, 23(5):30–37.
- Dowson, M.; Fernström, C.; (1994) “Towards Requirements for Enactment Mechanisms”. In: *European Workshop on Software Process Technology*, 3., 1994, Villard de Lans, France. Berlin: Springer-Verlag.
- Falbo, R.A., Arantes, D.O., Natali, A.C.C., (2004) “Integrating Knowledge Management and Groupware in a Software Development Environment”. *Proceedings of the 5th International Conference on Practical Aspects of Knowledge Management -PAKM'2004*, Karagiannis, pp. 94-105.
- Galotta, C.; Oliveira, K. M.; Rocha, A. R. C. (2004), “Apoio à Interação entre Processos de Negócio e de Software através de Gerência do Conhecimento”. *Simpósio Brasileiro de Qualidade de Software – SQBS*. Brasília, pp.
- Hansen, M. T.; Nohria, N.; Tierney, T. (1999) “What’s your strategy for managing knowledge”, *Harvard Business Review*, p. 106-116.
- ISO/IEC 12207 (2008) “ISO/IEC 12207:2008 Systems and Software Engineering - Software Life Cycle Processes”. Genebra: ISO, Março.
- Lima Reis, C. A.; Reis, R. Q. (2007) “Laboratório de Engenharia de Software e Inteligência Artificial: Construção do Ambiente WebAPSEE”. *ProQualiti – Qualidade na Produção de Software*. v. 3, n. 1, junho de 2007. p. 43-48.
- Montoni, M. A.; (2003) “Aquisição de Conhecimento: Uma Aplicação no Processo de Desenvolvimento de Software”. *Dissertação de Mestrado COPPE/UFRJ*.
- Natali, A. C., (2003) “Uma Infra-Estrutura para Gerência de Conhecimento em um Ambiente de Desenvolvimento de Software”. *Dissertação de Mestrado. UFES*.
- Natali, A. C. (2007) “Gerência de Conhecimento: Teoria e Prática”. *XXI Semana Paraense de Informática. III Congresso Paraense de Software*. Curso MPS.BR.
- Nonaka, I., Takeuchi, H. (1995) “The Knowledge-Creation Company: How Japanese Companies Create the Dynamics of Innovation”. *Oxford Univerty Press*, England.
- O’Really, T. (2005) “What is Web 2.0? Design Patterns and Business Models for the Next Generation of Software”, *International Journal of Digital Economics*, pp. 17-37.
- Probst, G. J. B.; Raub, S.; Romhardt, K. (1999) “Managing Knowledge: Building Blocks for Success”, 368 pp, Ed. Wiley.
- Rus, I.; Lindvall, M., (2002) “Knowledge management in software engineering”, *Software, IEEE*, vol.19, no.3, pp.26-38.
- Sales, E. O.; Reis, R. Q.; Freitas, Salomão F. (2006) “Uma Ferramenta para Recuperação de Modelos de Processo de Software Reutilizáveis.” *XIII Sessão de Ferramentas do SBES*. Florianópolis. p. 13-18.
- SOFTEX - Associação para Promoção da Excelência do Software Brasileiro (2007) “MPS.BR – Guia Geral”, versão 1.2, Junho. Disponível em: www.softex.br.