

Um Mecanismo de Suporte ao Gerenciamento de Recursos Humanos no Desenvolvimento Distribuído de Software

Elisa Hatsue Moriya Huzita, Tania Fatima Calvi Tait, Fabiana de Lima
{emhuzita, tait}@din.uem.br; fabiana.lima@brturbo.com.br

Universidade Estadual de Maringá
Avenida Colombo, 5790
Maringá- Estado Paraná-Brasil

Abstract

Software products resulting from large projects demand interaction among several activities, artifacts and people. Maintaining the control over software production and coordinating teams in this kind of project become extremely important and complex tasks. Products and processes quality improvement begin choosing the most capable individual to perform the tasks. When this choice is based on aspects of improvement process and capability model we believe it increases the possibilities of improving the quality of final product. The purpose of this paper is to present a mechanism which offers the project manager information on possible individuals whose profiles are more appropriated to perform a task. These information are generated according to rules based on aspects of improvement process and capability model, and provide decision support to project managers allowing human resource allocation for project activities.

Keywords: software distributed development, software project management, product and process quality, human resources management

Resumo

Os produtos de software resultantes de grandes projetos demandam a interação de muitas atividades, muitos artefatos e muitas pessoas. Nesse tipo de projeto, manter o controle sobre a produção do software e coordenar equipes tornam-se tarefas extremamente importantes e complexas. Manter e organizar a participação dos recursos humanos são tarefas que exigem um apoio considerável. A melhoria da qualidade de produtos e processos inicia-se com a escolha do indivíduo mais capacitado para a execução das atividades. Quando essa escolha está baseada processos de melhoria e modelos de capacitação, aumentam-se as possibilidades de melhoria na qualidade do produto final. Este artigo apresenta um mecanismo que oferece aos gerentes de projetos informações de pessoas que tenham o perfil mais adequado à atividade escolhida. Essas informações, que são geradas com base em regras cujas definições têm como fundamentação processos de melhoria e modelos de capacitação, auxiliarão de forma significativa os gerentes de projetos na tomada de decisão para a alocação dos recursos humanos.

Palavras-chave: Desenvolvimento distribuído de Software, Gerenciamento de Projeto de Software, Qualidade de produto e processo, Gerenciamento de Recursos Humanos.

1. Introdução

O atual panorama do desenvolvimento de software envolve projetos cada vez mais complexos, prazos de entrega cada vez menores e redução de custos. Além disso, tem se tornado cada vez mais frequente o desenvolvimento de software realizado por corporações geograficamente distribuídas, o que implica no envolvimento de pessoas cooperando na execução de atividades para atingir as metas estabelecidas.

A escolha adequada de indivíduo/equipe com melhor capacitação para a execução destas atividades podem contribuir para a melhoria da qualidade de produtos e processos.

Para ressaltar a importância de ferramentas de gerenciamento de projetos, basta reportar ao gerenciamento de recursos em um projeto (PMI, 2002). Manter e organizar uma grande quantidade de recursos, sejam eles logísticos, financeiros ou humanos são tarefas que exigem um esforço considerável. Para o gerenciamento de recursos logísticos e financeiros existe uma quantidade maior de ferramentas, como por exemplo, planilhas de cálculo e programas de apoio a recursos financeiros.

No entanto, o gerenciamento de recursos humanos ainda necessita de mais estudos devido às limitações da atual tecnologia disponível para coordenação de atividades. Segundo Reis (2003) ainda não se têm disponíveis mecanismos que ofereçam suporte na tomada de decisão referente à alocação de indivíduos para as atividades de projeto. A referida autora afirma ainda que as escolhas para tais alocações devem ser baseadas nas características e no histórico dos indivíduos.

O presente trabalho faz parte do projeto de pesquisa Uma Metodologia para Desenvolvimento Baseada em Objetos Distribuídos Inteligentes (Huzita, 2002). Este projeto tem por objetivo a construção de um ambiente distribuído de desenvolvimento de software, denominado DiSEN (*Distributed Software Engineering Environment*), o qual possui, entre outros recursos, uma ferramenta de gerenciamento de projetos denominada DIMANAGER (Huzita et al, 2004) .

No contexto do ambiente DiSEN, a ferramenta DIMANAGER trata do aspecto de acompanhamento no gerenciamento de projetos, possibilitando, por exemplo, o cadastro de projetos e atividades. Já o planejamento do projeto é realizado pelo gerente de projetos e/ou pelo coordenador de atividades, levando em conta seus conhecimentos sobre os indivíduos e as experiências acumuladas por eles, pautados em informações oriundas dos projetos finalizados ou em andamento.

Para aprimorar o processo de seleção de recursos humanos torna-se importante ter disponível um mecanismo que ofereça também suporte ao planejamento com informações a respeito das pessoas mais habilitadas para a execução de atividades no desenvolvimento distribuído de software. Para o projeto do mecanismo, levaram-se em consideração as características de *disponibilidade*, *habilidades* e *conhecimentos* dos indivíduos. Esse mecanismo apóia os gerentes de projetos na tomada de decisão na alocação de recursos humanos, fornecendo informações concisas e claras sobre as características e experiências dos indivíduos.

Este artigo está dividido nas seguintes seções: a metodologia de desenvolvimento da pesquisa é tratada na seção 2; o ambiente DiSEN bem como os *processos de melhoria (PSP e TSP)* e *modelo de capacitação (PCMM)* e outros trabalhos relacionados são discutidos na seção 3, a qual oferece uma visão do contexto onde está localizado essa pesquisa; o *mecanismo de apoio ao gerenciamento de recursos humanos desenvolvido* é apresentado na seção 4; a *validação do projeto*, na seção 5, contem os perfis dos grupos entrevistados para a validação do mecanismo implementado e a forma como essa validação foi realizada; e, finalmente, *conclusões e trabalhos futuros*, na seção 6.

2. Metodologia de Desenvolvimento da Pesquisa

A pesquisa desenvolveu-se em 3 fases: referencial teórico; projeto do mecanismo de suporte ao gerenciamento de recursos humanos e avaliação do mecanismo.

No item referencial teórico, realizou-se uma abordagem a partir do ambiente DiSEN (Huzita, 2002) no qual o trabalho está inserido; de ferramentas de gerenciamento de projetos (Celoxis, 2004); Gemetrics (Vavassori, 2002); DIMANAGER (Huzita et al, 2004) e também de modelos de processo e de produto (Humphrey, 1999).

A identificação das necessidades dos gerentes de projeto, quanto ao perfil desejado na alocação de recursos humanos, foi realizada por meio de pesquisa nas organizações e englobou entrevistas junto a profissionais liberais e em empresas.

Com base nos estudos e no levantamento efetuado, foi projetado o mecanismo que tem como fundamento a alocação de recursos humanos na atividade de planejamento de projetos. Após o projeto, procedeu-se à sua avaliação considerando os valores atribuídos a cada uma das características necessárias e de acordo com as regras definidas.

3. O Contexto do Mecanismo de Suporte ao Gerenciamento de Recursos Humanos

O mecanismo proposto foi elaborado a partir de três fontes: a falta de fornecimento de subsídios que facilitem a tomada de decisão por parte de gerentes de projetos na seleção de recursos humanos, mostrada em algumas estruturas de gerenciamento; o ambiente de software distribuído (DiSEN) e os Processos de Melhoria (PSP e TSP) e o Modelo de Capacitação (PCMM).

3.1 Trabalhos Relacionados

Os projetos GENESIS (Ballarini, 2003), o GROOVE (IKV, 2003), o ObjectWire (Weinrich, 1997), OPHELIA (Boldyreff, 2003), o PROSOFT (Reis, 2003); o Celoxis (Celoxis, 2004) e o GEMETRICS (Vavassori, 2002) foram analisados sob a ótica do fornecimento de apoio ao gerenciamento de recursos humanos.

Apesar das ferramentas GENESIS (Ballarini, 2003), o GROOVE (IKV, 2003), o ObjectWire (Weinrich, 1997), OPHELIA (Boldyreff, 2003) apresentarem características de gerenciamento de projetos, os aspectos relacionados com os recursos humanos, praticamente, não são tratados.

Existem alguns projetos como PROSOFT (Reis, 2003); o Celoxis (Celoxis, 2004) e o GEMETRICS (Vavassori, 2002) que oferecem suporte aos processos de software e que propõem o controle e, principalmente, a seleção dos recursos humanos em seus ambientes. Porém, esses projetos não fornecem implementações que automatizem o processo de escolha dos indivíduos.

Verificou-se, a partir desses projetos, a inexistência de implementações que realizem a avaliação dos recursos humanos considerando-se fatores relacionados ao conhecimento ou habilidades necessários à execução de atividades de projeto.

3.2. O Ambiente DiSEN

O projeto que originou o ambiente DiSEN busca suprir a crescente necessidade de apoio adequado ao processo de desenvolvimento distribuído de software. Essa necessidade foi causada pelo aumento da complexidade dos projetos atuais, o que dificulta o gerenciamento das atividades de construção de

software e de todo o cenário que as envolve, fazendo diminuir a qualidade dos produtos desenvolvidos. O estilo arquitetural do ambiente DiSEN é em camadas e baseia-se em agentes de software que devem ser desenvolvidos seguindo-se os padrões definidos pela *Foundation for Intelligent Physical Agents* (FIPA) (2002).

A primeira, é a camada Dinâmica, pela qual podem ser adicionados ou removidos do ambiente componentes ou serviços, enquanto ele estiver em execução. A segunda é a camada de Aplicação, na qual estarão contidos os gerenciadores e o repositório para persistência dos dados e metadados. A terceira é a de Infra-estrutura, que fornece o suporte às tarefas, como a persistência e o controle de concorrência. Nessa última camada está o canal de comunicação, por meio da qual é realizada a comunicação entre as camadas e também entre as ferramentas e o repositório (Huzita, 2002).

Como parte do ambiente DiSEN, na camada de aplicação foi definido o gerenciador de agentes, o qual contém agentes de software que auxiliam nas atividades do usuário, no trabalho cooperativo e na coleta de métricas do sistema. Os componentes do ambiente DiSEN devem garantir o trabalho cooperativo e as características de ambientes distribuídos, o que é suportado pelo gerenciador de *workspaces* que prove funcionalidades para que um ou mais espaços de trabalho cooperativos sejam criados e para que os usuários possam executar suas atividades com a certeza de que trabalhos realizados ou modificados serão persistidos no repositório central sempre que for requisitado ao ambiente.

Os *workspaces* são os espaços de trabalho, nos quais os usuários interagem, comunicam-se uns com outros compartilhando o mesmo espaço e tendo acesso às ferramentas e aos serviços (Huzita, 2002).

A MDSODI (Metodologia para Desenvolvimento de Software Distribuído) é uma das metodologias que poderão fazer parte do ambiente DiSEN e considera as características de sistemas distribuídos desde as fases iniciais do projeto. Nela foram especificadas as fases de: requisitos, análise, projeto e implementação, e testes e em cada uma delas foram definidos os passos a serem seguidos.

Para cada passo descrito na metodologia, uma ou mais atividades foram definidas, sendo que cada uma delas pode envolver uma ou mais tarefas. As atividades são o ponto de partida para a seleção que o gerente de projetos fará utilizando o mecanismo.

Na camada de aplicação do ambiente DiSEN é que se localiza o processo de gerência de projetos, a qual é apoiada pela ferramenta DIMANAGER (Huzita, 2002) para o planejamento e acompanhamento de projetos com a inclusão do mecanismo aqui proposto para seleção de recursos humanos no projeto.

3.3 Processos de Melhoria (PSP e TSP) e Modelo de Capacitação (PCMM)

A definição de fatores que sustentarão os pontos chaves considerados no mecanismo de suporte ao gerenciamento de recursos humanos baseou-se nos processos de melhoria *Personal Software Process* (PSP) (Humphrey, 1999), *Team Software Process* (TSP) (Humphrey, 1999) e no modelo de capacitação *People Capability Maturity Model* (PCMM) (Curtis et al, 2002) A adoção destes processos e modelo em empresas tem resultado em uma melhora significativa nos produtos e processos de software. Portanto, a utilização destes mostram-se adequadas para uma fundamentação sólida para o mecanismo ora apresentado.

- *Personal Software Process (PSP)*

Estabelece diretrizes ao indivíduo para que ele possa conhecer o próprio processo de trabalho e estabelecer a melhoria contínua sobre ele. Este processo surgiu pela necessidade de se aplicarem modelos de capacitação, tais como o CMM (*Capability Maturity Model*), em pequenos projetos de software, que até então eram aplicados para a capacitação de toda a organização.

O PSP está dividido em quatro níveis, que devem ser seguidos para que possa ser estabelecido por completo: medição pessoal, gerenciamento de pessoal, qualidade pessoal e processo cíclico pessoal. Esses níveis são divididos em atividades, como por exemplo, o planejamento de tarefas ou a medição de defeitos encontrados no software. Essas tarefas devem ser cumpridas para que o indivíduo estabeleça o próprio processo de melhoria contínua.

No processo pessoal são utilizadas várias técnicas, como por exemplo: gerenciamento e planejamento do tempo, gerenciamento de compromissos, planejamento e dimensionamento do produto, gerenciamento de prazos e planejamento do projeto. Essas podem ser adaptadas às necessidades de cada indivíduo.

- *Team Software Process (TSP)*

Estabelece as diretrizes para se alcançar a disciplina necessária na realização do trabalho em grupo e do seu gerenciamento.

O TSP é um processo que visa à constituição de grupos integrados, auxiliando projetistas na construção de equipes, na escolha do melhor processo a ser seguido, na reavaliação e no replanejamento periódicos do projeto. A estrutura do TSP é constituída de três grandes fases, que são: construindo habilidades (para disciplinar engenheiros), construindo grupos e trabalhando grupos.

A primeira fase é realizada com a utilização do PSP, para disciplinar os engenheiros em relação à medição e melhoria do processo pessoal de trabalho. A segunda fase visa disciplinar o grupo de trabalho, e tem o foco nas atividades de planejamento e a distribuição das atividades de projeto entre os membros do grupo. A terceira fase tem como foco disciplinar o gerenciamento. Assim, nela são estabelecidas atividades de monitoramento/acompanhamento do plano de projeto.

O presente trabalho foi desenvolvido com foco nas duas primeiras fases do processo, a de construção das habilidades individuais e a de grupos.

- *O People Capability Maturity Model (PCMM)*

É um modelo de capacitação pessoal que visa à melhoria da capacitação do indivíduo pertencente à organização em que é aplicado. Essa capacitação implica na melhoria dos conhecimentos, das habilidades e do processo que os envolve, bem como na cultura da organização.

O PCMM é constituído de cinco níveis: inicial, gerenciado, definido, previsível e otimizado. Cada um desses níveis possui áreas-chave de processos que são constituídas de práticas. Cada uma das áreas estabelecidas contém metas a serem atingidas, para que a organização possa atingir o próximo nível de maturidade, no processo de capacitação. As práticas estabelecidas na organização para alcançar um determinado nível de maturidade devem permanecer quando há a mudança de nível, para que a capacitação possa ser alcançada também no próximo nível. No que diz respeito a esse trabalho, o foco para o levantamento dos fatores considerados no mecanismo concentra-se nos níveis 2 e 3 desse modelo.

4. Apresentação de um Mecanismo de Suporte ao Gerenciamento de Recursos Humanos

O mecanismo desenvolvido teve como base os processos e o modelo descritos na seção 3 (PSP, TSP e PCMM), o ambiente DiSEN, as atividades definidas na MDSODI e as características de conhecimento, disponibilidade e habilidades.

No escopo desse trabalho, o termo *conhecimento* limita-se aos treinamentos dos quais o candidato participou ou ministrou, e às informações sobre conceitos que ele obteve por outros meios, e que o fizeram descrever o conceito de forma correta no questionário aplicado durante o desenvolvimento da pesquisa. No tocante à *disponibilidade*, foi relacionada à possibilidade do indivíduo ter em sua agenda horas suficientes para a execução de determinada atividade, no período em que ela deve ser executada. Em relação à *habilidade*, a avaliação do candidato foi realizada considerando-se a quantidade de projetos de software em que o indivíduo utilizou um conhecimento equivalente à habilidade. Para participação em projetos foi considerada a realização de tarefas pelo indivíduo, em quaisquer das fases de desenvolvimento (análise, projeto, codificação e teste).

4.1 As Funcionalidades do Mecanismo

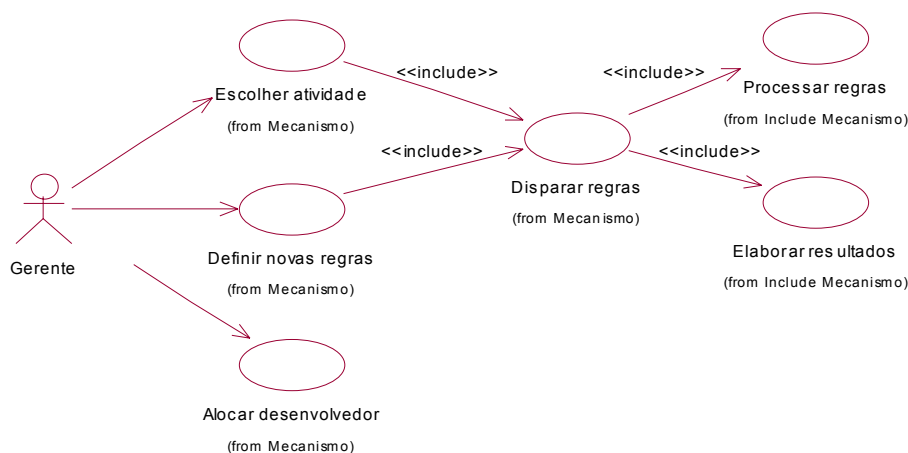


Figura 2. Diagrama de casos de uso do mecanismo

As funcionalidades identificadas para o mecanismo encontram-se ilustradas na Figura 2 e são descritas a seguir:

- *Escolher atividades*: representa a interface do mecanismo e o processo de disparo do agente de seleção, com os dados fornecidos pelo gerente, iniciando o processamento das regras referentes à fase ou atividade escolhida
- *Disparar regras*: cada atividade definida com base na metodologia MDSODI é constituída por um conjunto de regras, estabelecido através de regras *fuzzy*
- *Processar regras*: corresponde à busca e processamento das informações, pelo agente de busca, para a solução de determinada regra disparada
- *Elaborar resultados*: elabora uma solução com o agrupamento de todos os indivíduos disponíveis para seleção

Os casos de uso *definir novas regras* e *alocar desenvolvedor* não estão detalhados por estarem além do escopo deste trabalho.

4.2 A Arquitetura do Mecanismo

No projeto do mecanismo, ilustrado na figura 3, foram identificados dois agentes: o de busca de informações e o de seleção dos indivíduos mais aptos.

O agente de busca realiza a varredura das informações de todos os indivíduos em todos os nós do ambiente, e verifica quais desenvolvedores têm seu conhecimento ou sua habilidade classificado no nível requerido para a atividade escolhida pelo gerente. Ele permite que os dados do indivíduo sejam buscados em qualquer nó remoto pertencente ao ambiente, ou seja, todos os indivíduos que estiverem cadastrados no ambiente podem ser considerados na seleção.

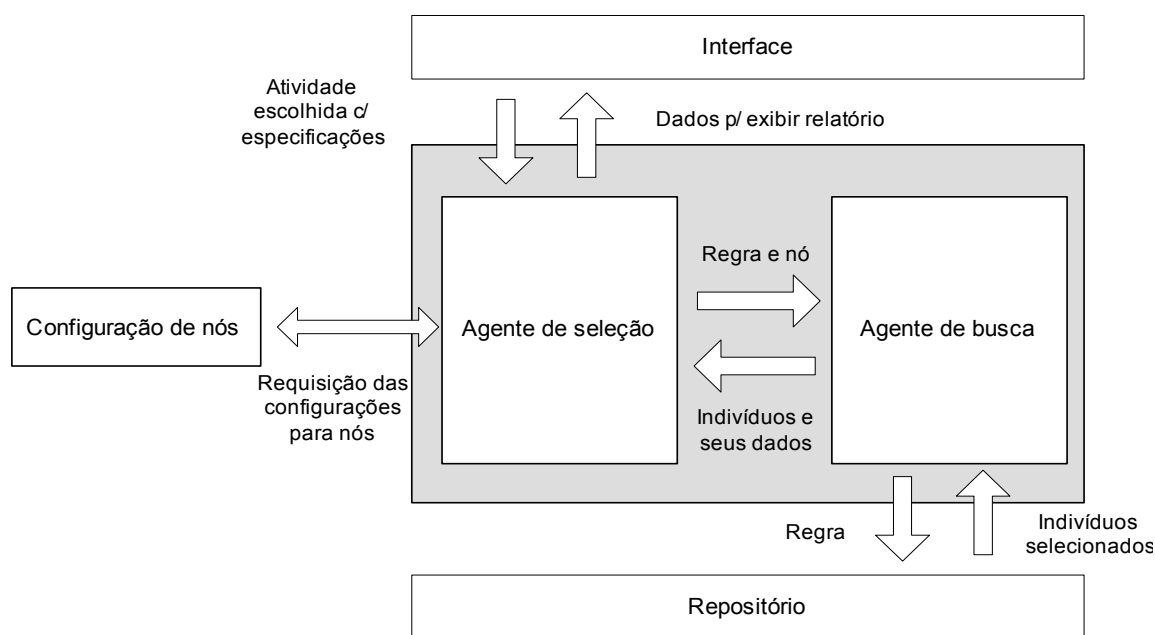


Figura 3. Arquitetura do Mecanismo de Suporte ao Gerenciamento de Recursos Humanos.

O agente de seleção executa as regras *fuzzy* e define, após essa execução, a classificação dos indivíduos em um relatório. O resultado final da avaliação é mostrado aos gerentes de projetos por esse relatório, que contém as informações sobre cada indivíduo e quais os conhecimentos ou habilidades avaliados sobre ele.

O mecanismo não realiza a escolha do indivíduo para os gerentes, mas indica as melhores opções para a atividade escolhida, para que eles escolham a opção que melhor lhes convier.

4.3 Definição das Regras

A execução de atividades previstas em um processo demanda conhecimentos e habilidades específicas a cada uma e a seleção adequada de quem vai executá-las determina fortemente a qualidade do produto final. Assim, foram selecionados fatores PSP, TSP e PCMM importantes a serem considerados para o estabelecimento de conhecimentos e habilidades necessários. O indivíduo selecionado para executar uma determinada tarefa deverá possuir o conhecimento e habilidades identificados. Um relacionamento

entre os fatores e as atividades através dos conhecimentos e das habilidades considerados como necessários é apresentado na Figura 4.

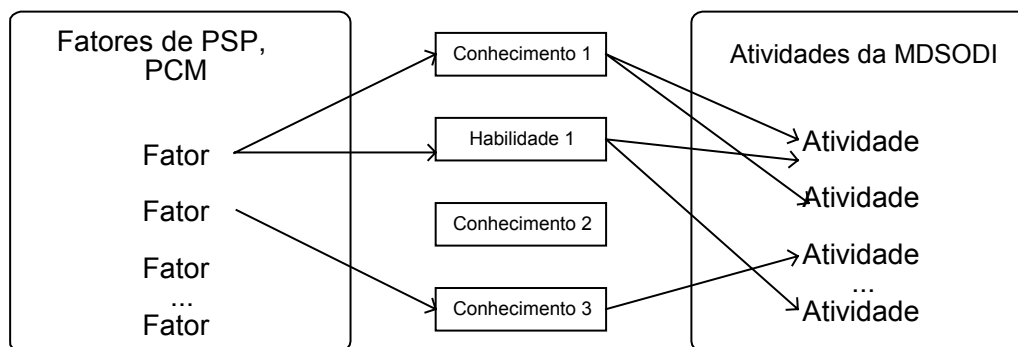


Figura 4. Relação Processos e modelo - conhecimentos/habilidades - Atividades

Foram elaboradas perguntas que possibilitassem avaliar o conhecimento ou a habilidade do indivíduo em linguagem de programação. Essas questões foram inseridas em um questionário que foi aplicado em dois grupos de organizações distintas e também com desenvolvedores que trabalham como autônomos, em projetos específicos (descrição na seção 5).

O conhecimento e habilidades necessárias são descritos como um conjunto de fatores que podem ser subdivididos em níveis para quantificá-los. Foram estabelecidos: muito alto, alto, médio, baixo e muito baixo. Para o seu cálculo utilizaram-se regras definidas com base nos conceitos da chamada lógica *fuzzy*, ou lógica nebulosa.

Para cada atividade são determinadas subdivisões de acordo com as habilidade e conhecimentos necessários. Para cada uma dessas subdivisões foram estabelecidos valores de acordo com a importância dada a cada item. Entrevistas realizadas com pessoas que desenvolviam a atividade no seu cotidiano forneceram informações para a elaboração dessas subdivisões.

A subdivisão descrita permite que a seleção de um candidato não seja baseada apenas em grupos com limitações rigorosas de avaliação (alto, médio ou baixo). Essa concepção de avaliação considera que um conhecimento ou uma habilidade não pode ser descrito apenas como bom ou ruim, alto ou baixo, um valor ou outro.

A seleção de um determinado candidato para a atividade escolhida pelo gerente de projetos realiza-se a partir de uma avaliação, resultante de uma classificação. Essa classificação qualifica o conhecimento ou a habilidade de um indivíduo utilizando os dados obtidos da execução de regras que efetuam cálculos com base em valores atribuídos a cada uma das partes em que cada um dos fatores tenha sido subdividido.

4.4 Tecnologias para a Implementação do Mecanismo

Na implementação do conjunto de fatores utilizou-se o FuzzyJ (2002) e JESS (2003) para as regras; JADE (Bellifemine, 2003) para os agentes. Todas as tecnologias citadas permitem a implementação e integração com a linguagem JAVA, que é a linguagem de programação adotada para os projetos ligados ao ambiente DiSEN.

No código de definição para conjuntos fuzzy (quadro 1), define-se a variável outLP e os conjuntos *fuzzy* que determinam os valores para o conhecimento em linguagem de programação. Em

L1, define-se uma variável do tipo *fuzzy*, cujos valores podem variar no intervalo [0,1], denominada outLP e definida através da grandeza de pontos. Logo abaixo, de L2 à L6, são definidos os intervalos *fuzzy* da variável, os chamados níveis de conhecimentos. Eles são: MB, B, M, A, MA.

```
// Linguagem de programação.  
L1 fuzzyvariable outLP = new fuzzyvariable ("LP", 0, 1, "pontos");  
L2 outLP addterm ("MB", new trianglefuzzzyset (0, 0.2, 0.35));  
L3 outLP addterm ("B", new trianglefuzzzyset (0.25, 0.4, 0.55));  
L4 outLP addterm ("M", new trianglefuzzzyset (0.45, 0.6, 0.75));  
L5 outLP addterm ("A", new trianglefuzzzyset (0.65, 0.8, 0.95));  
L6 outLP addterm ("MA", new Lfuzzzyset (0.85, 1));
```

Quadro 1 – Código de definição para conjuntos *fuzzy*.

5. Avaliação do Mecanismo de Gerenciamento de Recursos Humanos

Para a avaliação deste mecanismo foram realizadas entrevistas com desenvolvedores de software que trabalham em equipes e com desenvolvedores autônomos. Os entrevistados pertenciam a duas organizações distintas e seus perfis variam entre profissionais com mais de dez anos de atuação e recém-formados com experiência em desenvolvimento, com média de dois a três anos.

Nessas entrevistas utilizou-se um conjunto de questionários que tinham como objetivo verificar quais eram os conhecimentos, a disponibilidade e as habilidades dos participantes da entrevista, bem como uma visão geral de seu perfil e dos projetos dos quais eles participam ou já participaram. O conjunto de questionários elaborados incluiu informações pessoais, sobre projetos, sobre linguagem de implementação/prototipagem e conhecimentos e habilidades.

No final da aplicação dos questionários foi realizada uma avaliação por parte dos profissionais sobre o conteúdo dos questionários. Os profissionais participantes da pesquisa apontaram que: (1) as questões identificaram pontos sobre conhecimento e habilidades necessários em seu dia a dia ou em sua formação, que poderiam ser repensados ou que mereciam maior atenção; (2) puderam obter uma visão abrangente sobre os conhecimentos que são necessários para a execução do processo de desenvolvimento; (3) a abrangência das questões tornou o questionário extenso, apesar de necessário.

Finalmente, pode-se concluir que a avaliação realizada possibilitou a obtenção dos dados necessários para atingir os objetivos propostos pelo mecanismo aqui apresentado.

6. Conclusões e Trabalhos Futuros

Com o desenvolvimento do mecanismo de suporte ao gerenciamento de recursos humanos projetado deu-se um passo importante em relação ao apoio ao gerenciamento de projetos. Esse apoio permite que os gerentes tenham dados mais precisos para a tomada de decisão em relação à alocação de recursos humanos nas atividades do desenvolvimento de software.

O mecanismo proposto diferencia-se dos projetos anteriores por estar pautado nos processos de melhoria e no modelo de capacitação citados. Os processos e o modelo têm importância reconhecida perante as organizações que já implantaram partes ou todos os processos que os envolvem. Essas implantações indicam melhorias significativas no processo de desenvolvimento de software.

O resultado oferecido aos gerentes de projetos pelo mecanismo apresenta informações com as possíveis opções de pessoas que tenham o perfil mais adequado à atividade escolhida por eles, como parte de um processo de desenvolvimento de software. Essas informações são geradas com base nos

processos de melhoria e o modelo de capacitação, que irão auxiliar os gerentes de projetos na tomada de decisão para a alocação dos recursos humanos.

Com o intuito de aprimorar o mecanismo apresentado neste artigo mais alguns itens podem ser tratados: *(i)* construção de uma base de conhecimento contendo as características dos gerentes de projeto; *(ii)* desenvolvimento de um módulo de manutenção de regras; *(iii)* inserção de novos fatores relevantes no processo de seleção de recursos humanos para o desenvolvimento de software.

A construção de uma base de conhecimento contendo as características que gerentes de projetos experientes levam em conta, na alocação dos recursos disponíveis, em seus projetos trará um aprimoramento na atribuição de valores para os conhecimentos bem como na formulação das regras de seleção.

O desenvolvimento de um módulo de manutenção de regras que permita aos gerentes de projetos elaborar novas regras que seriam escolhidas como o melhor conjunto de conhecimentos/disponibilidade/habilidades a ser avaliado para um determinado caso torna-se possível a partir do mecanismo apresentado. No caso, pode ser incluída a abordagem de agentes inteligentes capazes de melhorar a classificação das regras escolhidas, devido a uma mudança na base de conhecimento estabelecida pelos gerentes de projetos.

Considerações sobre outros fatores relevantes para a seleção de recursos humanos, além de conhecimento, disponibilidade e habilidades, poderão vir a ser tratados e inseridos no mecanismo possibilitando a inserção de novos fatores, que passem a ser importantes no processo de desenvolvimento de software. Por exemplo, pode-se realizar um levantamento sobre fatores administrativos ou psicológicos relevantes para o processo de seleção. Esses fatores poderiam ser introduzidos como novas regras no mecanismo, desde que pudessem ser expressos computacionalmente, através de questões aplicáveis aos recursos disponíveis.

Salienta-se, finalmente, a integração do mecanismo de suporte ao gerenciamento de projetos na ferramenta DIMANAGER, o que contribui efetivamente para um planejamento adequado dos projetos e dos recursos humanos envolvidos, pelo aproveitamento das habilidades de cada participante no processo de desenvolvimento de software, o qual facilita a alocação de pessoas, também, no ambiente distribuído de software.

Referências

BALLARINI, D. et al . Modelling Real Requirements for Cooperative Software Development: A Case Study. IN: Workshop On Cooperative Support for Distriuted Software Engineering Processes, ., University of Sannio, Benevento. Advanced Program- Session 5. Itália. <<http://serg.ing.unisannio.it/esse2003/program.html>> Acesso em dezembro/2003

BELLIFEMINE, F. et al. JADE Programmer's Guide. JADE 3.1. 2003. 48f. TILAB S.p.A. 2003.

BOLDYREFF, C. et al. Environment to Support Colaborative Software Engineering. <<http://www.cs.put.pozm.an.pl/dweiss/site/publications>> Acesso em 3003.

CELOXIS Technologies Pvt. Ltd. Web Based Project Management Software Tool. Disponível em: <<http://www.celoxis.com>>. Acesso em: 12 jan. 2004.

CURTIS, Bill; REFLEY Bill; MILLER Sally. People Capability Maturity Model. Version 2.0. Technical Report CMU/SEI-2001-MM-001. 2001. Disponível em: <<http://www.sei.cmu.edu/publications/documents/01.reports/01mm001.html>>. Acesso em: 06 ago. 2003.

- FIPA Foundation for Intelligent Physical Agents. <http://www.fipa.org>. Acesso em agosto/2002.
- FUZZYJ Toolkit. Institute for Information Technology. 2002. Disponível em: <http://www.iit.nrc.ca/IR_public/fuzzy/fuzzyJToolkit.html>. Acesso em: 23 nov. 2003.
- HUMPHREY, Watts S. Pathways to process maturity: The personal software process and Team software process. Background, Volume 2, Issue 2. 1999. Disponível em: <<http://interactive.sei.cmu.edu/Features/1999/June/Background/Background.jun99.htm>>. Acesso em: 06 ago. 2003.
- HUZITA, E.H.M. Metodologia para desenvolvimento baseada em objetos distribuídos inteligentes. Relatório de Projeto de Pesquisa – Departamento de Informática. Maringá-PR, Universidade Estadual de Maringá, 2002.
- HUZITA, E.H.M; PEDRAS, M.E.V.; SANTIAGO, G. P; TAIT, T.F.C. DIMANAGER: a Tool for Distributed Software Development Management. Proceedings ICEIS 2004 – International Conference on Enterprise Information Systems, Porto, vol.3, pp.659-662.
- IKV Technologies AG. 2003 GROOVE Networks: Groove workspace. <<http://www.groove.net/products/workspace>> Acesso em novembro/2003
- JESS the Rule Engine for the Java™ Plataform. 2003. Disponível em: <<http://herzberg.ca.sandia.gov/jess/>>. Acesso em: 23 nov. 2003.
- PROJECT MANAGEMENT INSTITUTE (PMI). O universo do conhecimento em gerência de projetos – Brazil Minas Gerais Chapter. Belo Horizonte: 2002. Disponível em: <http://www.pmi.org/prod/groups/public/documents/info/pp_pmbokguide2000excerpts.pdf>. Acesso em: 06 ago. 2003.
- REIS C. A. L. Uma abordagem flexível para execução de processos de software evolutivos. 2003. 267 f. Tese (Doutorado) – Programa de pós-graduação em Ciência da computação, Instituto de informática, Universidade Federal do Rio Grande do Sul. Porto Alegre: 2003.
- VAVASSORI, F. B. Especificação do sistema. GEMETRICS v1.0. Laboratório de soluções em software. Centro tecnológico da terra e do mar/CTTMar. Universidade do Vale do Itajaí. Santa Catarina: 2002. Disponível em: <<http://inf.univali.br/~gemetrics/esp.htm>>. Acesso em: 06 ago. 2003.
- WEINRICH, R. ALTMANN, J. An Object-oriented Infrastructure for a Cooperative Software Development Environment. Proceedings...International Symposium on Applied Corporate Computing (ISACC-97), 10 p, November 5-7, Montrey, México, ITESM, 1997.