

Uma abordagem Baseada em Agentes de Apoio à Gestão de Projetos de Software

José Ferdinandy Silva Chagas¹, Francisco Milton Mendes Neto¹, Mariela Inés Cortés²

¹Universidade Federal Rural do Semi-Árido (UFERSA)
Departamento de Ciências Exatas e Naturais – Mossoró, RN – Brasil

²Universidade Estadual do Ceará (UECE)
Departamento de Estatística e Computação – Fortaleza, CE – Brasil

{ferdinandy,miltonmendes}@ufersa.edu.br, mariela@larces.uece.br

Abstract. *Software process improvement models can contribute significantly in product quality and evolution of organizations. However, implement these models is a major challenge. This process generates an increase in cognitive load to project manager that must manage projects following the recommendations of model. This paper proposes a multi-agent web system to assist in knowledge management and automation tasks recommended by CMMI (Capability Maturity Model Integration) Level 2.*

Resumo. *Modelos de melhoria de processo de software podem contribuir significativamente na qualidade do produto e evolução das organizações. No entanto, a implementação destes modelos é um grande desafio. Esse processo gera um aumento na carga cognitiva do gerente de projeto que deve gerenciar os projetos seguindo as recomendações do modelo. Este artigo propõe um sistema web multi-agente para auxiliar na gestão do conhecimento e automação de tarefas recomendadas pelo modelo CMMI (Capability Maturity Model Integration) Nível 2.*

1. Introdução

A implantação de modelos de melhoria de processos cria desafios para a organização, principalmente por inserir passos adicionais que sobrecarregam o gerente de projeto, além de existir possíveis conflitos entre a cultura organizacional e as recomendações do modelo. O desafio torna-se maior quando a organização em questão é de pequeno ou médio porte, pois não possui os recursos financeiros e humanos para incentivar e executar a implantação do modelo de melhoria do processo de software de modo adequado [Furucho and Aguiar 2012].

O processo de software constitui um processo intensivo em conhecimento, envolvendo diversos perfis, que manipulam uma gama de informações. A adoção de uma solução que satisfaça todas as necessidades de conhecimento relacionadas aos processos da Engenharia de Software (ES) não é uma tarefa trivial. Isto se torna mais difícil quando a organização adota um modelo de qualidade de processos, como o CMMI (*Capability Maturity Model Integration*) [Chrissis et al. 2011]. Apesar dos investimentos por parte de instituições de pesquisa e Organizações de Desenvolvimento de Software (ODS) em

ambientes automatizados de suporte a processos de software, não se tem alcançado a qualidade desejada e a produtividade esperada da equipe. Na ES, a experiência de trabalho, que ajuda a evitar erros do passado e melhorar decisões futuras, ainda reside, principalmente, nos colaboradores da ODS.

O trabalho apresentado em [Huang and Zhang 2010] aborda os problemas que surgem quando o CMMI é implantado em pequenas e médias empresas. Um dos problemas que o artigo aponta é que o processo de implantação do CMMI requer muitos recursos financeiros, humanos e tecnológicos. Além disso, causa um aumento na carga de trabalho e isto leva a um relativo aumento na expropriação de recursos. Outro fator importante que afeta a melhoria dos processos é a falta de pessoal especializado dedicado à qualidade dos processos.

Segundo Huang e Zhang [Huang and Zhang 2010], o CMMI apresenta uma média de 600 (seiscentas) práticas entre gerais e específicas. Todas essas práticas precisam ser planejadas, monitoradas, medidas, analisadas e validadas de acordo com o gerenciamento do projeto. Este é um processo muito complexo para pequenas e médias empresas. Assim, é muito importante utilizar ferramentas de suporte ao CMMI durante o processo de implementação do modelo. Tais ferramentas podem contribuir para minimizar a carga cognitiva do gerente de projeto que, além de precisar monitorar todos os aspectos relativos à execução do projeto em si, precisará também ter conhecimento das práticas relativas ao CMMI. Este trabalho pretende contribuir com a gestão de projetos de software propondo uma solução automatizada de apoio ao gerente projeto seguindo as especificações das áreas de processo do CMMI-Nível 2. Esse nível propõe os requisitos básicos para o correto gerenciamento de um projeto de software. O objetivo deste trabalho é auxiliar na implantação e manutenção do Nível 2 do CMMI, reduzindo a carga cognitiva do gerente de projeto. A ferramenta proposta, denominada de VPM (*Virtual Project Manager*), consiste em um sistema web suportado por agentes de software que monitoram e recomendam tarefas seguindo as práticas do CMMI com base nos processos do PMBOK (*Project Management Body of Knowledge*) [ANSI and PMI 2008], além de notificar o gerente de projeto quanto a outras metas que os agentes não podem controlar.

O trabalho proposto envolve a integração de agentes de suporte ao CMMI a uma ferramenta de gerenciamento de projetos. Essa integração visa auxiliar na Gestão de Conhecimento [Davenport and Prusak 1998] e realizar a automação de algumas tarefas recomendadas nas áreas de processo do CMMI para o Nível 2. Esse nível é considerado um grande desafio para implantação do CMMI visto que é o marco inicial. As tarefas apoiadas pelo VPM estão relacionadas com as seguintes áreas: gerenciamento de requisitos; monitoramento e controle do projeto; e planejamento do projeto.

Para prover suporte automatizado à implantação do CMMI, é utilizada a tecnologia de agentes de software. Para representação do conhecimento que é utilizado pelos agentes do sistema são utilizadas ontologias. Esse modelo de conhecimento será utilizado pelos agentes que auxiliarão o gerente de projeto na gestão do conhecimento relacionado com o projeto. Este trabalho está organizado da seguinte forma: na Seção 2, são descritos os principais conceitos abordados; a Seção 3 apresenta uma explanação sobre os trabalhos relacionados; a Seção 4 descreve o processo de modelagem e desenvolvimento do sistema multiagente para gestão do conhecimento em projetos de software; na Seção 5, são apresentados cenários de uso, como exemplos de utilização do sistema; por fim, na

Seção 6, são apresentadas as considerações finais e os trabalhos futuros.

2. Referencial Teórico

A seguir uma breve contextualização sobre os conceitos envolvidos no desenvolvimento do trabalho.

2.1. CMMI

O CMMI é um modelo evolutivo criado pelo Instituto de Engenharia de Software (*SEI - Software Engineering Institute*) para melhoria de processos organizacionais. O modelo no momento da edição deste trabalho está na versão 1.3 e é composto por 3 (três) modelos principais: o CMMI-ACQ (*CMMI for Acquisition*), CMMI-DEV (*CMMI for Development*) e CMMI-SVC (*CMMI for Services*). O modelo foco do trabalho é o CMMI-DEV, que é formado por 22 (vinte e duas) áreas de processos, cada uma com suas metas genéricas e específicas [Chrissis et al. 2011].

Existem duas abordagens de implantação do CMMI: a abordagem por estágios e a abordagem contínua. A abordagem por estágios abrange as metas genéricas de várias áreas de processos, enquanto a abordagem contínua foca apenas em algumas metas específicas de cada área. As organizações que cumprem com as exigências do modelo seguindo a abordagem por estágio ou contínua recebem uma certificação CMMI-DEV com base no seu nível de maturidade ou capacidade, respectivamente. O foco deste trabalho é o Nível 2 de maturidade do CMMI-DEV. Nesse nível, a organização se dedica à melhoria dos processos básicos que se não forem realizados resultarão em problemas para o projeto e para a organização. Segundo o SEI [SEI 2012], entre os anos de 2009 a 2011, aproximadamente 636 (seiscentas e trinta e seis) empresas alcançaram o Nível 2 do CMMI.

O modelo CMMI-DEV constitui uma referência para melhoria de processos, porém não enfatiza detalhes sobre como as organizações devem proceder para atingir tais requisitos. Isto ocorre pela variação das disciplinas envolvidas nos processos de cada ODS. Contudo, o PMBOK pode ser utilizado como referência para preencher essa lacuna e detalhar técnicas e procedimentos úteis para cumprir com o modelo CMMI [Ehsan et al. 2010]. Assim, este trabalho utiliza o PMBOK como guia para detalhar as atividades necessárias para melhoria dos processos das organizações.

2.2. Sistemas Multiagentes

Segundo Russel e Norvig [Russell and Norvig 2010], um agente é uma entidade que pode perceber o seu ambiente por meio de sensores e agir sobre ele por meio de atuadores. Agentes de software são programas automatizados que executam ações de forma autônoma e, se necessário, se comunicam com outros agentes a fim de atingir os seus objetivos.

Os sistemas computacionais podem utilizar um ou mais agentes de software para melhor atender às suas necessidades. Quando o sistema trabalha com um conjunto de agentes que interagem entre si, ele é denominado de sistema multiagente (SMA). Um SMA é formado por um conjunto de agentes que podem trabalhar de forma cooperativa ou competitiva. As principais características que diferenciam um agente de software de

um programa comum são sua autonomia e a capacidade de socialização para alcançar objetivos [Russell and Norvig 2010].

Entre os tipos de agentes mais comuns temos os agentes reativos simples e os agentes reativos baseados em modelo. O agente reativo simples representa apenas comportamentos de ação e reação, sem guardar nenhum registro de suas ações. Agentes baseados em modelos são agentes que, além de reagirem sobre um ambiente, possuem uma representação interna de conhecimento, que será considerada na ponderação de suas ações [Russell and Norvig 2010]. O trabalho apresentado faz uso desses dois tipos de agentes.

2.3. Ontologias

Gruber [Gruber et al. 1993] define ontologia como uma especificação explícita de uma conceitualização. No caso de sistemas baseados em conhecimento podemos descrever a ontologia como um conjunto de termos que representam um domínio. O conjunto de objetos que são representados por esses termos formam o universo do discurso. Os objetos representados e os relacionamentos existentes entre eles são formalmente descritos através de um vocabulário representativo. Assim, através deste vocabulário um programa de computador pode representar o conhecimento.

3. Trabalhos Relacionados

A seguir serão apresentados trabalhos que abordam o desafio da implantação de modelos de melhoria de processos discutido neste trabalho e como a solução proposta se diferencia das demais.

Em [Zhang and Shao 2011], os autores propõem um *framework* para melhoria de processos de pequenas e médias organizações de software. O objetivo é auxiliar essas organizações nos estágios iniciais do seu desenvolvimento focando na qualidade dos seus produtos. O *framework* é um guia que estrutura todo o processo de desenvolvimento do projeto de software seguindo as recomendações das áreas de processos do CMMI para os níveis 2 e 3. O trabalho apresenta a importância e um referencial básico sobre o modelo CMMI, e propõe um *framework* como guia prático. O *framework* proposto no trabalho é um guia para ser aplicado durante o desenvolvimento de projetos.

Em [Garcia et al. 2010], os autores propõem uma ferramenta web para auxiliar na melhoria de processos nas organizações de software implantando o CMMI. No trabalho foi realizada uma pesquisa sobre as características mais comuns em ferramentas de apoio à melhoria de processos. Os autores enfatizam que existem poucos trabalhos sobre desenvolvimento de ferramentas SPI (*Software Process Improvement*). A maioria das ferramentas explora o processo de avaliação baseando-se em padrões. A ferramenta apresentada propõe um guia para melhoria dos processos das organizações seguindo modelos como o CMMI. O sistema disponibiliza recursos para acompanhamento e monitoramento do processo de melhoria facilitando a compreensão e a integração com os processos da organização.

Kovacheva e Todorov [Kovacheva and Todorov 2011] propõem o uso da ferramenta baseada na web TRAC [Team 2012] para integrar o modelo CMMI e as práticas ágeis. Segundo os autores, os fatores mais importantes para o sucesso da integração são: automação de processos; cumprimento da gestão; motivação da equipe; e documentação

do processo. A ferramenta TRAC suporta todos esses critérios aumentando a motivação da equipe, fornecendo estimativas mais precisas, documentação detalhada o suficiente, e baixos custos de mudanças.

Em [Homchuenchom et al. 2011], os autores apresentam a ferramenta SPIALS para auto-avaliação de processos em organizações de software. A aplicação implementa as recomendações do CMMI mapeando as recomendações para práticas do SCRUM [Schwaber and Beedle 2002]. O SCRUM é uma metodologia ágil utilizada para desenvolvimento ágil de software. A ferramenta proposta realiza uma auto-avaliação dos processos de uma organização utilizando o método SCAMPI [Team 2011]. Este é o método padrão utilizado pelo SEI para avaliação de maturidade e capacidade de acordo com o CMMI.

Os trabalhos destacados discutem os problemas que ocorrem na implantação de modelos de melhoria de processos de software e como solução são propostos sistemas de apoio. Ferramentas que apoiem a melhoria dos processos podem contribuir de forma significativa para melhoria na qualidade dos produtos, principalmente, de pequenas e médias empresas de software. Essas ferramentas podem reduzir custos ao passo que promovem a qualidade dos produtos.

A ferramenta VPM, proposta pelo presente trabalho, se diferencia dos trabalhos apresentados por guiar o desenvolvimento de projetos seguindo as recomendações de melhoria de processos de forma proativa. As ferramentas propostas nos trabalhos relacionados exploram principalmente a auto-avaliação de processos das organizações. O processo de auto-avaliação consiste na comparação entre os processos organizacionais e as recomendações do CMMI. Assim, elas funcionam como guias no caminho da implantação do modelo. Porém, soluções como estas criam passos adicionais para o gerente de projeto sem auxiliar diretamente nas tarefas de gerenciamento. A VPM disponibiliza recursos automatizados para gerenciar projetos de software, de forma proativa em tempo de execução, contemplando as práticas do CMMI.

4. VPM

Para apoiar a implantação do Nível 2 do CMMI, foram identificadas as áreas de processos vinculadas a esse nível de maturidade. Na abordagem por estágios, para ser atingido um determinado nível é necessário cumprir todas as metas genéricas de todas as áreas de processos para aquele nível. As áreas de processos para o Nível 2 do CMMI são:

- Gerenciamento de Configuração;
- Medição e Análise;
- Monitoramento e Controle do Projeto;
- Planejamento do Projeto;
- Garantia de Qualidade de Processos e Produtos;
- Gerenciamento de Requisitos;
- Gerenciamento de Acordo com Fornecedores.

A partir de uma análise mais detalhada de cada área de processo, foram identificadas quais áreas seriam apoiadas através da ferramenta. Com base nesse estudo, as áreas de processo escolhidas foram: gerenciamento de requisitos, planejamento do projeto, e monitoramento e controle do projeto. As outras áreas de processos pertencentes

ao Nível 2 e suas respectivas práticas envolvem procedimentos interpessoais, ou são áreas com difícil abstração para uma solução genérica. Procedimentos interpessoais são os que envolvem a interação entre os membros da equipe e os stakeholders. As áreas de difícil abstração são áreas muito específicas, cujas soluções estão diretamente ligadas ao perfil do projeto.

Essas características dificultariam ou limitariam a implementação de uma solução genérica automatizada. A tentativa de automatizar tais processos poderia criar passos adicionais desnecessários. Por exemplo, a área de processo de Gerenciamento de Acordo com Fornecedores envolve contato com fornecedores, definição de acordos, revisão de documentos e outras tarefas manuais.

O modelo CMMI é formado por um conjunto de recomendações que indicam o que deve ser feito, porém não enfatiza como deve ser feito. As ferramentas propostas nos trabalhos relacionados focam na avaliação dos processos para que se alinhem às recomendações. O processo de auto-avaliação das organizações e a implantação do CMMI de forma geral implicam na realização de passos adicionais que aumentam a carga cognitiva do gerente de projeto. A ferramenta apresentada neste trabalho propõe um SMA para recomendar e automatizar algumas tarefas de gerenciamento seguindo as recomendações do CMMI. A idéia é reduzir a carga cognitiva do gerente de projeto durante e após a implantação do CMMI.

Para a automação das tarefas são utilizadas as tecnologias de agentes de software e ontologias. Os agentes auxiliam de forma proativa no monitoramento dos processos, enquanto as ontologias são utilizadas como bases de conhecimento flexíveis e como entrada para mecanismos de inferência. Para cada área de processo selecionada foram atribuídos agentes. Assim, os principais agentes do sistema são: Assistente de Requisitos, Assistente de Plano e Assistente de Monitoramento e Controle. Além desses, temos um Assistente de Interface, responsável pela comunicação com o gerente de projeto e com a equipe. A partir das áreas de processo selecionadas foram identificadas metas, práticas e subpráticas de cada área de processo e cada subprática foi associada a um objetivo de um agente.

Na Tabela 1 são apresentadas as práticas e subpráticas contempladas por cada agente da ferramenta.

4.1. Arquitetura da Aplicação

Para evitar o retrabalho de implementar funções já características de ferramentas de GPS, foi selecionada uma ferramenta open-source e modular, chamada dotProject [Jordan 2008], onde foi adicionado o módulo VPM. Assim, foi criado um módulo para o dotProject que possui acesso a todas as informações dos projetos cadastrados e dispõe das funcionalidades do dotProject no que diz respeito às funções gerais de gestão. Enquanto as informações são gerenciadas através da ferramenta dotProject, o SMA monitora a base de dados e as ações do gerente de projeto.

Os agentes foram desenvolvidos através do framework JADE (*Java Agent DEvelopment*) [Bellifemine et al. 2007]. Este framework disponibiliza um conjunto de bibliotecas Java que facilitam o desenvolvimento de agentes seguindo as especificações da FIPA (*Foundation for Intelligent Physical Agents*) [FIPA 2000]. A FIPA é uma organização da IEEE Computer Society que define padrões para tecnologia de agentes e promove a interoperabilidade de seus padrões com outras tecnologias. Além disso, a

Tabela 1. Práticas e subpráticas apoiadas pela VPM

Assistente	Prática	Subprática
Requisitos	Gerenciar Requisitos	Entender requisitos; Gerenciar mudanças de requisitos; Manter a rastreabilidade bidirecional dos requisitos; Garantir o alinhamento entre artefatos e requisitos.
Plano	Estabelecer Estimativas.	Estimar esforço e custo.
	Desenvolver o Plano do Projeto.	Identificar riscos do projeto; Planejar recursos do projeto; Planejar as necessidades de conhecimento e habilidades.
Monitoramento e Controle	Monitorar o Projeto em Relação ao Plano	Monitorar riscos do projeto.
Interface	Monitorar o Projeto em Relação ao Plano	Conduzir revisões de progresso.

ferramenta JADE possui uma plataforma própria para execução dos agentes e interface gráfica para gerenciamento.

As atividades previstas no PMBOK estão especificadas na ontologia utilizada pelo ambiente. As ontologias foram desenvolvidas em OWL [Dickinson 2012], através da ferramenta Protégé [Knublauch et al. 2004] como em [Gualberto et al. 2012], e são acessadas pelo SMA através da API Java Apache Jena [Dickinson 2012]. A ontologia utilizada pelo Assistente de Requisitos é a Ontologia de Atividades. Ela armazena as atividades e subatividades de acordo com o PMBOK, além de informações sobre os membros da equipe e as partes interessadas.

A arquitetura da aplicação é apresentada na Figura 1. O gerente de projeto interage com a ferramenta através da interface da aplicação web de gestão de projetos, que, por sua vez, utiliza um banco de dados. Os agentes interagem com o dotProject e monitoram o banco utilizado pela ferramenta. Os agentes são o Assistente de Plano (AP), o Assistente de Monitoramento e Controle (AMC), o Assistente de Requisitos (AR) e o Assistente de Interface (AI), que interage diretamente com o gerente de projeto e a equipe através da interface da ferramenta ou por e-mail. Os agentes AP e AR possuem bases de conhecimento representadas em ontologias. Cada base de conhecimento está diretamente relacionada com os objetivos do agente. O ambiente possui três ontologias: Ontologia de Riscos (OR), Ontologia de Atividades (OA) e Ontologia de Recursos Humanos (ORH).

Além dos agentes já citados, a arquitetura também apresenta o agente DF (*Directory Facilitator*), que é o agente responsável por disponibilizar publicamente os serviços dos demais agentes. Esse mecanismo permite que os agentes se comuniquem. Esse tipo de abordagem segue as especificações da FIPA [FIPA 2000], que define um padrão a ser utilizado para arquiteturas de sistemas multiagente.

4.2. Modelagem dos Agentes

Seguindo as práticas apresentadas na Tabela 1, foram modelados os agentes utilizando a metodologia MAS-CommonKADS+ [Oliveira 2010]. Um dos principais agentes do sistema é o Assistente de Requisitos. O seu Modelo de Agente é apresentado na Figura 2.

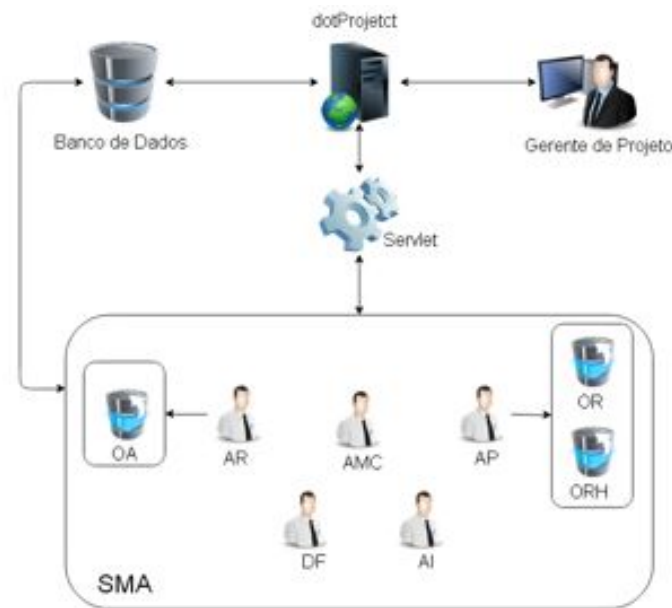


Figura 1. Arquitetura do Ambiente

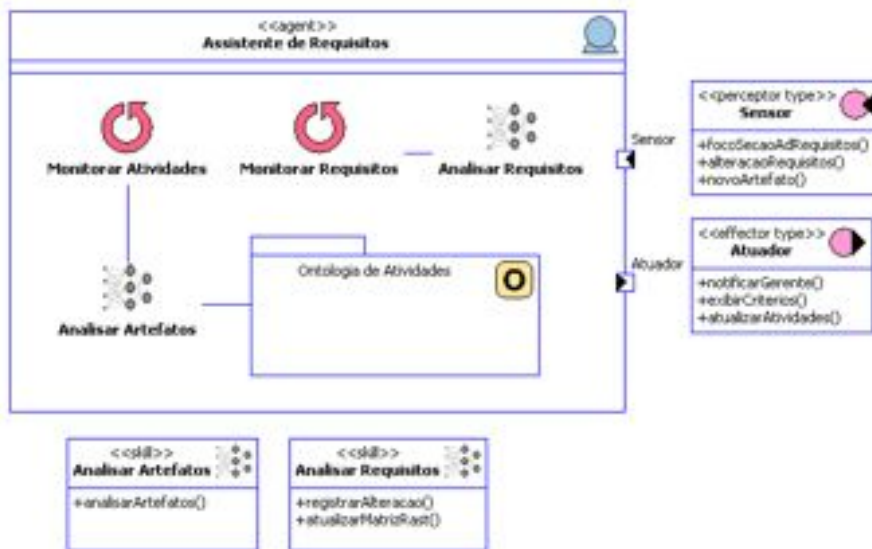


Figura 2. Modelo do Agente Assistente de Requisitos

Como pode ser observado na Figura 2, este agente possui dois comportamentos: Monitorar Requisitos e Monitorar Atividades. O comportamento de Monitorar Requisitos possui os seguintes objetivos: (i) auxiliar no entendimento dos requisitos; (ii) registrar alterações dos requisitos; e (iii) manter a rastreabilidade dos requisitos. O comportamento de Monitorar Atividades tem o objetivo de fazer o correto sequenciamento de atividades com base nos artefatos necessários para iniciá-las. Assim, só serão executadas atividades cujas necessidades de artefatos sejam completadas. Essas necessidades e a dependência das atividades estão especificadas na Ontologia de Atividades utilizada pelo agente.

5. Cenários de Uso

Nesta seção segue uma descrição de cenário para a funcionalidade de gerenciamento de atividades da ferramenta VPM. O objetivo é apresentar de forma contextualizada a utilização das tecnologias e das abordagens do projeto.

As funcionalidades representadas pelos modelos de sequência nas Figuras 3 e 4 contemplam uma das recomendações do CMMI. Essa recomendação visa o correto alinhamento entre artefatos e requisitos. Segundo a recomendação, as atividades e artefatos devem ser revisados para que sigam o plano do projeto. O responsável por cumprir com esse objetivo é o agente Assistente de Requisitos (AR).

O modelo de sequência é baseado em um conjunto de passos realizados para cumprir determinada tarefa, além de mostrar quais atores e recursos estão envolvidos.

5.1. Cenário 1 - Atualizar Projetos

No cenário descrito na Figura 3 é apresentado o processo que ocorre quando um novo projeto passa a ser acompanhado pela VPM.

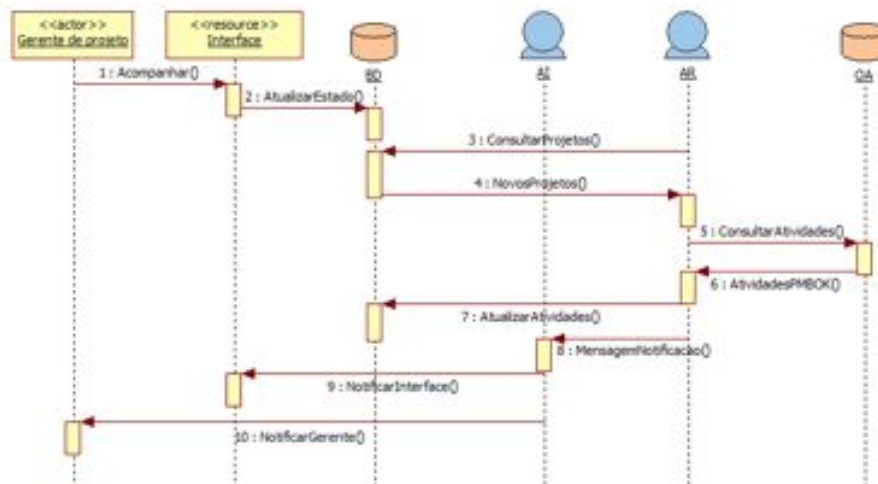


Figura 3. Modelo de sequência para o Cenário 1

O gerente de projeto, utilizando a ferramenta através da interface web, define que um determinado projeto será acompanhado pela VPM (1: Acompanhar()). A interface, após a ação do gerente de projeto, atualiza o estado do projeto no banco de dados (2: AtualizarEstado()). Constantemente o agente AR irá consultar o banco de dados do dot-Project em busca de novos projetos acompanhados (3: ConsultarProjetos()). O banco de dados retornará ao agente AR a lista de todos os projetos atualmente acompanhados (4: NovosProjetos()). Quando o agente AR identifica um novo projeto acompanhado, ele consulta a Ontologia de Atividades (5: ConsultarAtividades()). Essa consulta resulta na lista atualizada das atividades e subatividades que devem ser associadas aos projetos de acordo com o PMBOK (6: AtividadesPMBOK()). Quando o agente AR carrega as atividades da Ontologia de Atividades, ele então atualiza a lista de atividades dos projetos em acompanhamento pelo VPM (7: AtualizarAtividades()). Após atualizar as atividades

o agente AR envia uma mensagem de notificação ao agente AI (8: MensagemNotificacao()). O agente AI, ao receber a notificação do agente AR, notifica o gerente de projeto e a equipe via Interface (9: NotificarInterface()). Em seguida o agente AI notifica o gerente de projeto via e-mail (10: NotificarGerente()).

5.2. Cenário 2 - Analisar Artefatos

No cenário na Figura 4 é apresentado o processo que ocorre quando novos artefatos são associados aos projetos monitorados. São comparados os artefatos contemplados e os artefatos necessários para realização das tarefas.

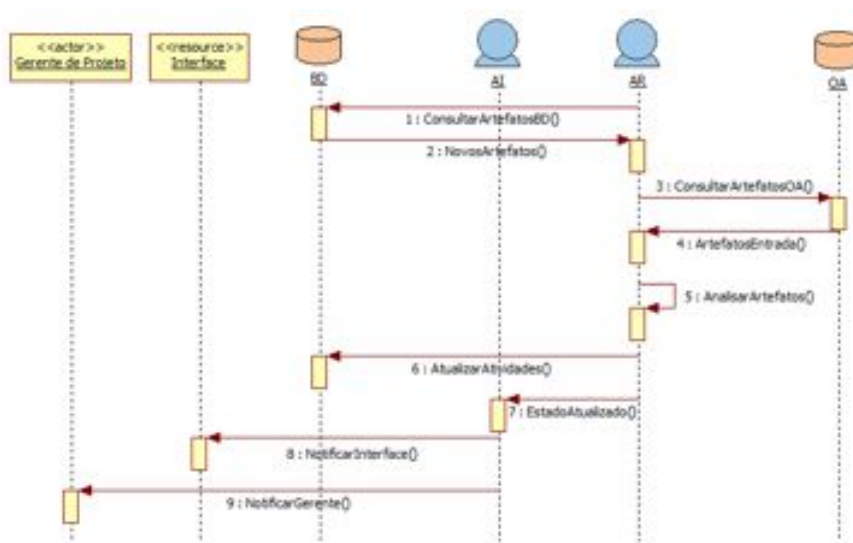


Figura 4. Modelo de sequência para o Cenário 2

O agente AR monitora o banco de dados do dotProject em busca de novos artefatos associados aos projetos monitorados (1: ConsultarArtefatosBD()). O banco de dados retorna a lista completa de artefatos associados aos projetos (2: NovosArtefatos()). Ao verificar que novos artefatos foram associados aos projetos monitorados o agente AR consulta na ontologia de atividades. O agente busca todos os artefatos necessários para entrada das atividades associadas aos projetos (3: ConsultarArtefatosOA()). A consulta realizada pelo agente AR retorna a lista de artefatos necessários para entrada das atividades (4: ArtefatosEntrada()). Com a lista de novos artefatos associados aos projetos e a lista de artefatos carregados da ontologia de atividades, o agente compara as listas. A partir dessas informações o agente identifica quais atividades possuem o conjunto de artefatos de entrada completo e quais não possuem (5: AnalisarArtefatos()). Após realizar a análise dos artefatos o agente AR atualiza o estado das atividades dos projetos monitorados. As atividades cujos artefatos de entrada já foram contemplados estão liberadas para execução, as que não estão são informadas as pendências (6: AtualizarAtividades()). Quando o AR atualiza as atividades dos projetos ele notifica o AI (7: EstadoAtualizado()). Ao perceber que os estados das atividades foram atualizados o AI notifica a Interface (8: NotificarInterface()). Além de notificar os usuários através da interface o agente AI notifica o gerente de projeto via e-mail (9: NotificarGerente()).

6. Considerações Finais e Trabalhos Futuros

Neste artigo foi apresentado um sistema multiagente para gestão de conhecimento em projetos de software com o propósito de diminuir a sobrecarga cognitiva sobre o gerente de projetos de ODS na implantação, ou manutenção, do Nível 2 do CMMI. A solução proposta se diferencia das demais abordagens citadas na literatura por evitar a criação de passos adicionais ao processo de gestão utilizando agentes de software e ontologias. Essas tecnologias podem ser utilizadas para automatizar as tarefas de forma flexível e apoiar o trabalho do gerente de projetos de forma proativa durante a execução das tarefas de gerenciamento.

Esse tipo de abordagem para ferramentas de gerenciamento de projetos poderá facilitar o trabalho de implantação de modelos de melhoria e no desenvolvimento de produtos com maior qualidade. Como trabalhos futuros, pretende-se melhorar a arquitetura SMA proposta, adicionando novos agentes para contemplar outros níveis do CMMI, além de melhorar os agentes existentes em relação ao uso de outras técnicas inteligentes visando o aprimoramento de suas atividades. Pretende-se ainda tornar a arquitetura proposta genérica o suficiente para ser utilizada com outras ferramentas de gerenciamento de projetos e realizar um estudo de caso em uma ODS para validar na prática a eficácia do sistema proposto.

Referências

- ANSI and PMI (2008). *A Guide to the Project Management Body of Knowledge: PMBOK Guide*. Project Management Institute, Inc.
- Bellifemine, F., Caire, G., and Greenwood, D. (2007). *Developing multi-agent systems with JADE*, volume 5. Wiley.
- Chrissis, M., Konrad, M., and Shrum, S. (2011). *CMMI® for Development: Guidelines for Process Integration and Product Improvement*. Addison-Wesley Professional.
- Davenport, T. and Prusak, L. (1998). *Conhecimento empresarial*. Rio de Janeiro: Campus.
- Dickinson, I. (2012). The jena ontology api. Disponível em: <http://jena.sourceforge.net/ontology/index.html>. Acessado em: Abril 2012.
- Ehsan, N., Malik, O., Shabbir, F., Mirza, E., and Bhatti, M. (2010). Comparative study for pmbok & cmmi frameworks and identifying possibilities for integrating itil for addressing needs of it service industry. In *Management of Innovation and Technology (ICMIT), 2010 IEEE International Conference on*, pages 113–116. IEEE.
- FIPA, F. (2000). Agent management specification. Disponível em: <http://www.fipa.org/>. Acessado em: Abril 2012.
- Furucho, R. A. and Aguiar, M. A. (2012). A importância do cmmi para o sucesso das organizações. *Engenharia de Software Magazine*, (46):17–25.
- Garcia, I., Pacheco, C., and Calvo-Manzano, J. (2010). Using a web-based tool to define and implement software process improvement initiatives in a small industrial setting. *Software, IET*, 4(4):237–251.
- Gruber, T. et al. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220.

- Gualberto, E. S., de Sousa Jr, R. T., de Deus, F. E., and Duque, C. G. (2012). Infosecrm: Uma abordagem ontológica para a gestão de riscos de segurança da informação. In *VIII Simpósio Brasileiro de Sistemas de Informação (SBSI 2012). Trilhas Técnicas*.
- Homchuenchom, D., Piyabunditkul, C., Lichter, H., and Anwar, T. (2011). Spials: A light-weight software process improvement self-assessment tool. In *Software Engineering (MySEC), 2011 5th Malaysian Conference in*, pages 195–199. IEEE.
- Huang, D. and Zhang, W. (2010). Cmmi in medium & small enterprises: Problems and solutions. In *Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference on*, pages 171–174. IEEE.
- Jordan, L. (2008). *Gerenciamento de projetos com dotProject: guia de instalação, configuração, customização e administração do dotProject*. Pearson Prentice Hall.
- Knublauch, H., Ferguson, R., Noy, N., and Musen, M. (2004). The protégé owl plugin: An open development environment for semantic web applications. *The Semantic Web- ISWC 2004*, pages 229–243.
- Kovacheva, T. and Todorov, N. (2011). Optimizing software development process: A case study for integrated agile-cmmi process model. In *EUROCON-International Conference on Computer as a Tool (EUROCON), 2011 IEEE*, pages 1–2. IEEE.
- Oliveira, M. J. (2010). *Mas-commonkads+: Uma extensão a metodologia mas-commonkads para suporte ao projeto detalhado de sistemas multiagentes racionais*.
- Russell, S. and Norvig, P. (2010). *Artificial intelligence: a modern approach*. Prentice hall.
- Schwaber, K. and Beedle, M. (2002). *Agile software development with Scrum*, volume 18. Prentice Hall PTR Upper Saddle River^ eNJ NJ.
- SEI (2012). Published CMMI appraisal results. Disponível em: <http://sas.sei.cmu.edu/pars/>. Acessado em: Abril 2012.
- Team, S. (2011). Standard cmmi appraisal method for process improvement (scampi) a, version 1.3: Method definition document.
- Team, T. (2012). The trac user and administration guide. Disponível em: <http://trac.edgewall.org/wiki/TracGuide>. Acessado em: Dezembro 2012.
- Zhang, L. and Shao, D. (2011). Software process improvement for small and medium organizations based on cmmi. In *Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on*, pages 2402–2405. IEEE.