Spider-PE: Uma Ferramenta de Apoio à Execução de Processos de Software aderente ao CMMI-DEV e MR-MPS

Antônio A. C. Silva¹, Elder J. F. Silva¹, Carlos S. Portela², Alexandre M. L. Vasconcelos², Sandro R. B. Oliveira¹

¹Programa e Pós-Graduação em Ciência da Computação e Faculdade de Computação – Instituto de Ciências Exatas e Naturais (ICEN) – Universidade Federal do Pará (UFPA) Caixa Postal 479 – 66.075-110 – Belém – PA – Brasil

²Centro de Informática (CIn) – Universidade Federal de Pernambuco (UFPE) Caixa Postal 7851 – 50.740-560 – Recife – PE – Brasil

{aandrecunhas, elderferreirass}@gmail.com, {csp3, amlv}@cin.ufpe.br, srbo@ufpa.br

Abstract. This paper presents a free software tool, called Spider-PE, which aims to support the software process enactment in a flexible and semi-automated way. This tool is one of the results of the SPIDER Project research, which provides technology solutions to meet best practice described in CMMI-DEV and MR-MPS quality models. It is expected that this tool could facilitate the enactment monitoring of the software process and reduce the costs of this phase. This cost reduction occurs due to the use of free technologies aiming no costs to the organization.

Resumo. Este artigo apresenta uma ferramenta de software livre, denominada Spider-PE, que objetiva apoiar a execução do processo de software de forma flexível e semi-automatizada. Esta ferramenta é resultado de pesquisas do Projeto SPIDER, o qual apresenta soluções tecnológicas a fim de atender as boas práticas descritas nos modelos de qualidade CMMI-DEV e MR-MPS. Espera-se que esta ferramenta possa facilitar o acompanhamento da execução do processo de software e reduzir os custos desta fase devido a proposta utilizar somente tecnologias livres, sem custos para a organização.

1. Introdução

Desenvolver software tem se mostrado uma atividade bastante complexa, pois nem todas as organizações desenvolvedoras conseguem atingir o escopo definido, assegurar o cumprimento de prazos, estimar corretamente custos e recursos ou garantir que o software desenvolvido possua a qualidade esperada pelo usuário [The Standish Group International, 2009]. Com o objetivo de solucionar estes problemas, a Engenharia de Software tem produzido ferramentas de auxílio ao desenvolvimento de software, assim como estudos sobre formas de controle do processo de desenvolvimento [Reis, 2003].

Segundo Oliveira *et al.* (2005), uma das evoluções mais importantes no estudo da qualidade de software está na constatação de que a qualidade do processo de produção do software é tão importante quanto a qualidade do produto final. A partir

deste conhecimento, surgiram importantes mecanismos de avaliação e certificação de qualidade de software baseados na maturidade e capacidade das organizações desenvolvedoras na condução de seus processos. Assim, modelos internacionais como o CMMI-DEV (*Capability Maturity Model Integration for Development*) [SEI, 2010] e modelos nacionais como o MR-MPS (Modelo de Referência para Melhoria do Processo de Software) [SOFTEX, 2011], vêm sendo amplamente utilizados para avaliar os processos de desenvolvimento de software.

Neste contexto, o objetivo deste artigo é apresentar a ferramenta Spider-PE (*Process Enactment*), uma solução de software livre de apoio à execução flexível e semi-automatizada de processos de software, aderente aos modelos de qualidade CMMI-DEV e MR-MPS. A ferramenta é parte integrante do Projeto SPIDER (*Software Process Improvement: DEvelopment and Research*) [Oliveira *et al.*, 2011], desenvolvido na Universidade Federal do Pará. Este projeto tem como um dos focos principais apresentar soluções tecnológicas (ferramentas de software livre, *frameworks*) com características adequadas para atender as boas práticas descritas nestes modelos de qualidade. A iniciativa do projeto SPIDER em adotar tecnologias livres para o desenvolvimento de ferramentas de software livre justifica-se em um movimento crescente de esforços individuais e coletivos, impulsionado pela massificação da Internet, para a criação de softwares não proprietários que objetivam atender necessidades comuns de organizações e indivíduos [GNU, 2012].

Além desta seção introdutória, a Seção 2 deste artigo discorre sobre as etapas de definição, melhoria e execução do processo de software. Esta seção também trata dos trabalhos relacionados e da abordagem do Projeto SPIDER. Na Seção 3 é descrita a ferramenta Spider-PE, a partir dos seus objetivos e características, tecnologias utilizadas e seus módulos e funcionalidades. A Seção 4 trata dos resultados esperados deste trabalho, e por fim, a Seção 5 apresenta as considerações finais.

2. Processos de Software: Definição, Execução e Melhoria

Um processo de software descreve o conjunto de tarefas necessárias para transformar os requisitos dos usuários em software [Humphrey, 1989]. Neste sentido, o processo de software pode ser definido como um roteiro que descreve metodologias, ferramentas e mecanismos que possuem a finalidade de auxiliar no desenvolvimento de uma solução automatizada que atenda o escopo, o tempo e a qualidade esperada pelo cliente [Pressman, 2006]. A seguir, são apresentados os principais conceitos e características do processo de software.

2.1. Conceitos e Características

Uma descrição simplificada do processo de software pode ser encontrada no seu modelo, que se caracteriza como uma abstração de um processo real que contêm as atividades, os papéis e os produtos (artefatos) [Pressman, 2010]. Este modelo de processo de software descreve o desenvolvimento de software de maneira a indicar quem, quando, onde, como e por que os passos são realizados [Lomchamp, 1993].

Para representar os elementos que compõem o processo, ou seja, para construir modelos de processo de software, é necessária uma linguagem para modelagem do processo [Feiler e Humphrey, 1993]. De um modo geral, estas linguagens para modelagem de processo estabelecem uma representação das atividades e tarefas do processo de software e os recursos relacionados com essas atividades e tarefas.

Depois de modelado, o processo entra na etapa de execução, onde será executado, controlado, validado e aperfeiçoado em curtos ciclos incrementais e iterativos [Bendraou *et al.*, 2007]. De acordo com Reis (2003), esta fase de execução dependerá de um mecanismo automatizado que compreenda o processo modelado e oriente os desenvolvedores no decorrer do seu trabalho, assim como execute automaticamente algumas tarefas.

A fim de produzir softwares com qualidade e se manterem competitivas no mercado, as organizações desenvolvedoras de software devem ser capazes de melhorar continuamente os seus processos de software [SOFTEX, 2011]. Esta melhoria contínua pode ser facilitada quando estes processos são orientados por padrões e modelos de referência de processos. Baseado nesta constatação, dois modelos de melhoria são considerados neste artigo: o CMMI-DEV [SEI, 2010] e o MR-MPS [SOFTEX, 2011].

2.2. Trabalhos Relacionados

Segundo Reis (2000), a tecnologia de processo de software fundamenta-se na descrição formal do processo, que objetiva a sua automação, verificação, aperfeiçoamento e reutilização. Nas pesquisas feitas a respeito desta tecnologia, observa-se o uso de ferramentas CASE (*Computer-Aided Software Engineering*) e de ambientes responsáveis por integrá-las, os ADSs (Ambientes de Desenvolvimento de Software). Assim, surgiram os PSEEs (*Process-Centered Software Engineering Environments*) a partir da necessidade de se incorporar aos ADS mecanismos que permitem a descrição do processo e acompanhamento das atividades pelo ambiente. Um dos componentes deste PSEE é a Máquina de Execução, responsável por executar ou interpretar o modelo de processo de acordo com o especificado.

A seguir são descritas características de PSEEs identificados na literatura especializada, com foco no módulo de execução e no apoio à avaliação e melhoria de processos de software. Posteriormente, serão apresentados pontos fortes, pontos fracos e oportunidades de melhoria destas abordagens.

O WebAPSEE [Lima et al., 2006] é um ambiente de gerência de processos baseado em software livre, criado através da parceria entre várias instituições, destacando-se o Laboratório de Engenharia de Software da Universidade Federal do Pará (LabES/UFPA) e a Regional Belém do Serviço Federal de Processamento de Dados (SERPRO). Utiliza uma linguagem visual própria para modelagem, a WebAPSEE-PML (*Process Modeling Language*) que se baseia na especificação formal e abordagens de gramáticas de grafos definida em [Reis, 2003]. França et al. (2009) destaca o uso desta ferramenta na adoção do MPS.BR nível G, numa organização desenvolvedora de software. Contudo, não há evidências explícitas do atendimento de alguns resultados deste nível, como, a rastreabilidade de dados.

O ODE (Ontology-based software Development Environment) é um ambiente concebido pelo Laboratório de Engenharia de Software da Universidade Federal do

Espírito Santo (LabES/UFES), tendo por base uma ontologia específica para Qualidade de Software [Guizzardi *et al.*, 2008]. O Projeto ODE justifica que ADSs construídos com base em ontologias permitem uma integração mais fácil com diferentes ferramentas que apoiam atividades de engenharia de software correlacionadas.

A Estação TABA [Travassos, 1994] procura abranger ambientes de desenvolvimentos de software apropriados às especificidades das organizações, processos de software e projetos. Possui duas ferramentas relacionadas à Avaliação e Melhoria de Processos: AvalPro e Pilot. A ferramenta AvalPro apoia as atividades do grupo de Garantia da Qualidade do Processo e do Produto (GQPP) da organização. Já a Pilot apoia a avaliação de propostas de melhoria de um processo de forma sistemática, planejada e controlada. Através destas ferramentas, a Estação TABA apresenta o suporte explícito a áreas de processo do CMMI 2 e 3 e níveis G a C do MPS.BR.

O *ImPPros* [Oliveira, 2007] é um ambiente que aborda a melhoria contínua do processo de software. Essa melhoria tem como base a transformação do processo de software com base nos possíveis mapeamentos entre os modelos e normas da qualidade.

O Quadro 1 resume as características destes ambientes para execução e melhoria de processos de software, destacando seus pontos fortes, fracos e oportunidades de melhoria.

Quadro 1 - Comparação entre Características de PSEEs

| Ambientes | Pontos Fortes | Pontos Fracos | Oportunidades de Melhoria |
|-----------------|---|--|--|
| WebAPSEE | Auxilia a adoção de Modelos de Qualidade; Baseado em software livre; Possui parceiros na indústria de software. | Possui linguagem própria; Não explicita formalmente a aderência a modelos de qualidade. | Mapeamento e conversão entre a sua linguagem e as linguagens padrões da OMG. |
| ODE | Baseado em ontologias; Possui uma ontologia específica para qualidade. | Não adota linguagem de modelagem. | Relacionar a proposta com linguagens padrões da OMG e modelos de qualidade. |
| Estação TABA | Abrange diversas áreas do conhecimento; Atendimento ao MPS.BR e CMMI. | Adota linguagem própria; Não possui flexibilidade. | Mapeamento e conversão entre a sua linguagem e as linguagens padrões da OMG. |
| ImPProS | Leva em consideração o uso de modelos e normas da qualidade que orientem melhoria contínua do processo. | A execução de processo é apresentada de forma textual. | Inserir formalismo de execução de processos. |

Nota-se, em geral, que a maioria destas ferramentas adota linguagem própria. Geralmente, estas linguagens são concebidas sem considerar boas práticas recomendadas pelos modelos de qualidade. Os diferenciais do trabalho apresentado neste artigo, em relação aos mapeados no Quadro 1, são melhor detalhados na Seção 4.

2.3. Abordagem do Projeto SPIDER

A abordagem do projeto SPIDER para ciclo de vida de processo estabelece 5 (cinco) fases: Definição do Processo, Simulação do Processo, Execução do Processo, Avaliação e Melhoria do Processo. Esta abordagem é apresentada na Figura 1.

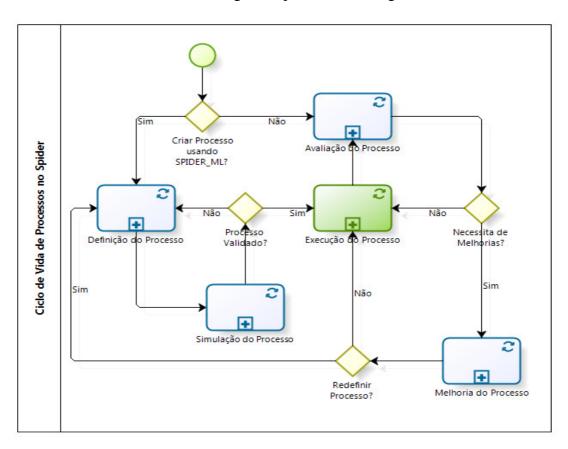


Figura 1 - Ciclo de Vida de Processos no contexto do Projeto SPIDER

Observa-se, de acordo com este ciclo de vida, a necessidade inicial de se modelar um processo de software. Se a organização já possuir um processo definido, o próximo passo será avaliar o processo já estabelecido, de forma a identificar se este processo necessita ou não de melhorias. Se necessária a melhoria do processo, este poderá ser redefinido na fase de Definição. Caso o processo não necessite de melhorias, a fase de Execução pode ser iniciada.

Se, ao início do ciclo, a organização não possuir um processo estabelecido, verifica-se a necessidade de definição de um processo a partir da linguagem de modelagem SPIDER_ML [Barros e Oliveira, 2010a], que se caracteriza como um perfil do padrão SPEM (*Software Process Engineering Metamodel Specification*), e por meio do apoio da Spider-PM [Barros e Oliveira, 2010b], uma ferramenta que permite a modelagem de processos a partir das notações desta linguagem.

Então, a partir do processo definido, é possível realizar uma simulação do processo, onde a finalidade é identificar possíveis problemas na definição do processo antes que este seja executado. Caso o processo não tenha sido validado na simulação, este deve ser redefinido. Caso contrário, o processo pode ser executado.

Nota-se, portanto, que todas as fases convergem para a fase de Execução do Processo, que é a fase responsável pelo auxílio na coordenação das atividades realizadas por pessoas (gerentes e equipe de desenvolvimento) e por ferramentas automatizadas.

No que diz respeito à fase de Execução, definiu-se no Projeto SPIDER a linguagem de execução de processos xSPIDER_ML (*eXecutable SPIDER_ML*) [Portela *et al.*, 2012], que se caracteriza como uma extensão da linguagem de modelagem SPIDER_ML e que permite a execução dos processos modelados a partir das notações do SPEM. Em paralelo, realizou-se um mapeamento entre os modelos de qualidade CMMI-DEV e MR-MPS [Portela e Gomes, 2011a], a fim de identificar as recomendações destes modelos para a fase de Execução.

Por fim, estabeleceu-se um *framework* denominado Spider-PE (*Process Enactament*) [Portela e Gomes, 2011a], que objetiva definir um fluxo conceitual de atividades genéricas para que as organizações desenvolvedoras de software possam executar seus processos de software de forma semi-automatizada e de acordo com as recomendações dos modelos de qualidade CMMI-DEV e MR-MPS. O *framework* estabelecido organiza-se em 3 (três) fases: Fase de Gerenciamento do Processo, onde acontece o planejamento do processo e são definidas as atividades de monitoramento e controle; Fase de Execução das Atividades do Processo e Fase de Aplicação do Formalismo de Execução, que são responsáveis pela aplicação do formalismo de execução da xSPIDER_ML na execução das atividades do processo.

Destaca-se, portanto, a necessidade do uso de uma Máquina de Execução que adote o formalismo de execução proposto neste trabalho a fim de realizar a automatização das atividades do *framework*. A próxima seção apresenta a ferramenta de software livre Spider-PE, foco deste artigo, que incorpora uma Máquina de Execução aderente às atividades deste *framework*.

3. A Ferramenta Spider-PE

Conforme descrito na Subseção 2.2, a maioria das abordagens para a execução de processos de software não adotam práticas dos modelos de qualidade. Buscando uma alternativa que atendesse aos principais padrões adotados pela indústria de software, propôs-se o desenvolvimento de uma ferramenta, que objetiva prover funcionalidades como planejamento, monitoramento, controle e comunicação, dentre outras atividades relacionadas à execução do processo.

A Spider-PE é uma ferramenta de licença GPL – *General Public License* [GNU, 2012] voltada especificamente para apoiar a abordagem apresentada na Subseção 2.3, a qual traz uma solução para semi-automatizar a execução de processos de software. Esta ferramenta foi concebida como um ambiente *desktop* desenvolvido na linguagem Java e pautado no uso de tecnologias livres, como: o IDE (*Integrated Development Environment*) Eclipse 3.7, o SGBD (Sistema Gerenciador de Banco de Dados) MySQL 5.5; o *framework* de mapeamento objeto-relacional Hibernate 4.0; a biblioteca de

serialização de objetos para XML e vice-versa, *XStream*; a biblioteca para criação e manipulação de arquivos PDF, *iText*; e a biblioteca para desenho de diagramas baseados na teoria dos grafos *JGraph*.

3.1. Ferramentas Integradas

Além das tecnologias livres usadas no desenvolvimento da ferramenta, descritas no início desta seção, também foram usadas ferramentas de software livre, consolidadas no mercado, que apóiam atividades da Spider-PE. Estas tecnologias são descritas a seguir.

Para apoio à gestão de mudança é usado o *SVN* (*Subversion*), sistema que adota a Licença Apache de software livre, disponível em *subversion.apache.org*. Na Spider-PE utiliza-se este para gerenciar a evolução do projeto através do controle e acesso de versões dos produtos de trabalho. Também se utiliza o *Redmine*, ferramenta de licença GPL que, no contexto das funcionalidades desta ferramenta, cadastra, monitora e acompanha a solução de possíveis problemas que possam surgir durante a execução do projeto. Esta ferramenta encontra-se disponível em *www.redmine.org*.

No apoio às estimativas de processo, adotam-se ferramentas de licença GPL desenvolvidas pelo Projeto SPIDER¹: *Spider-APF*, que permite realizar estimativas utilizando o método de análise por pontos de função; *Spider-UCP*, utilizada para medição de tamanho de software a partir do método Pontos por Caso de Uso (*Use Case Points – UCP*); e a *Spider-CoCoMo*, que provê apoio à estimativa de prazo, esforço e tamanho de equipe utilizando o método CoCoMo - *Constructive Cost Model*.

A avaliação de processo é feita através da ferramenta *Spider-CL*, que também adota a licença GPL. Esta ferramenta objetiva o desenvolvimento de *checklists*, o uso destes em diferentes contextos e o registro destes usos. No contexto da Spider-PE, utiliza-se a *Spider-CL* como forma de avaliar os produtos de trabalho gerados e as áreas do processo através de critérios objetivos (descritos em *checklists* de avaliação).

E no apoio ao processo de medição, utiliza-se a *Spider-MPlan*, licença GPL, que possibilita a definição, coleta, análise e acompanhamento de medidas. Na Spider-PE, esta ferramenta é utilizada na definição de métodos para monitorar e controlar as atividades que estão sendo desenvolvidas durante o projeto, a fim de garantir que a realização destas ocorra conforme o planejado.

O uso das ferramentas apresentadas nesta subseção permite a reutilização de funcionalidades que atendam as recomendações dos modelos de qualidade CMMI-DEV e MR-MPS, evitando o desenvolvimento de funcionalidades específicas para atender determinadas atividades do *framework* Spider-PE, descrito na Subseção 2.3.

3.2. Módulos e Funcionalidades

O *framework* para execução de processos, descrito na Subseção 2.3, define um modelo que apoia o gerenciamento da execução de processos de software. Nesse contexto, a ferramenta Spider-PE foi desenvolvida a fim de realizar a automatização das atividades deste *framework*, sendo tais atividades estabelecidas a partir do mapeamento dos

¹ As ferramentas desenvolvidas pelo Projeto SPIDER, sob licença GPL, encontram-se disponíveis em *spider.ufpa.br/index.php?id=resultados*.

modelos de qualidade MR-MPS e CMMI-DEV e do formalismo da linguagem xSPIDER_ML. Para cada atividade especificada no *framework* existe uma funcionalidade correspondente na ferramenta, tornando, desta forma, a ferramenta aderente aos modelos de melhoria de qualidade do processo de software.

A ferramenta Spider-PE utiliza o conceito de módulos para especificar um determinado conjunto de funcionalidades, agrupadas de acordo com as fases do framework e dos atores responsáveis por sua execução. Sendo assim, esta ferramenta possui 3 (três) módulos: Administração, Gerência do Processo e Execução do Processo. O primeiro módulo a ser acessado é o de Administração, responsável pela configuração das ferramentas descritas na Subseção 3.1. Este módulo também é responsável pela conversão do arquivo XML com a modelagem do processo (construída na Spider-PM através das notações da linguagem SPIDER_ML) para o esquema de banco de dados relacional. Este processo de conversão do arquivo XML para o banco de dados relacional é realizado com o auxílio da biblioteca XStream, que provê a serialização de objetos para XML e vice-versa. O módulo de Administração também permite definir um usuário Gerente de Processo para o processo a ser executado.

Depois de definido, o *Gerente de Processo* pode acessar o módulo de *Gerência do Processo*, apresentado na Figura 2. Este módulo baseia-se na aderência as boas práticas dos modelos de qualidade MR-MPS e CMMI-DEV, por isto, está relacionado às atividades de planejamento e monitoramento do processo. O módulo de *Gerência do Processo* é composto de três etapas: Planejamento, Execução e Monitoramento. Este módulo inicia-se a partir do Planejamento do processo, onde o *Gerente de Processo* tem acesso a diversas funcionalidades, conforme mostra Figura 2, detalhadas a seguir:

- **Política Organizacional**: esta atividade consiste em definir e publicar uma política organizacional que estabeleça diretrizes e responsabilidades para áreas específicas do processo, como Gerência de Projeto e Gerência de Requisitos;
- Estimativa de Esforço: consiste em estimar o esforço necessário para realizar as atividades do processo (apoio ferramental da *Spider-UCP*, *APF* e *CoCoMo*);
- Cronograma de Tarefas: baseada na estimativa de esforço da atividade anterior, essa atividade consiste em definir o cronograma do processo;
- **Riscos Identificados**: consiste em identificar os riscos do projeto e definir um plano de mitigação e contingência para estes riscos. Dependendo do impacto do risco, pode haver a necessidade de reestimar o esforço;
- Capacidade e Treinamento: esta atividade consiste, a partir da análise entre as habilidades necessárias para realizar as atividades do processo, em identificar as capacidades dos membros da Equipe do Projeto. Ainda nesta etapa, verifica-se a necessidade de realização de treinamentos para capacitação da equipe;
- Recursos e Responsabilidades: consiste em definir os recursos (hardware e software) e as responsabilidades (recursos humanos) necessários para realizar as atividades do processo. Depois de identificados, realiza-se a alocação destes recursos para o projeto;
- Plano de Comunicação: consiste em identificar os interessados (*stakeholders*) afetados pela realização do projeto, a fim de planejar a comunicação entre estes;

- Avaliação dos Produtos de Trabalho: identificar os requisitos dos produtos de trabalho, a partir da definição de padrões, *templates* e demais atributos necessários para a geração de cada produto de trabalho (apoio da *Spider-CL*);
- Avaliação do Processo: esta atividade consiste em gerar um *checklist* para a avaliação das diversas áreas do processo, como Requisitos, Análise e Projeto, dentre outras (apoio da *Spider-CL*);
- **Definição de Medição**: define as atividades de monitoramento e controle, a fim de garantir que a realização do processo ocorra conforme o planejado (apoio ferramental da *Spider-MPlan*).

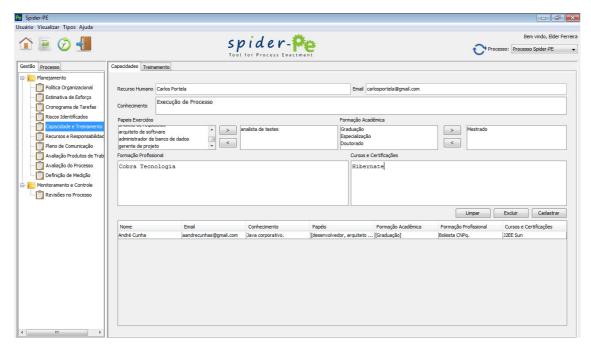


Figura 2. Módulo de Gerência do Processo

Uma vez a etapa de Planejamento devidamente realizada, será iniciada a execução do processo, onde o *Gerente de Projeto* deve gerenciar a comunicação durante o andamento do projeto. A Equipe do Projeto, por sua vez, deve executar as atividades do processo. Esta etapa corresponde ao módulo de *Execução do Processo* que será acessado pelo *Usuário do Processo* e será detalhada posteriormente.

Por fim, o módulo de *Gerência do Processo* possui a etapa de Monitoramento, onde o *Gerente de Projeto* monitora os problemas que podem ocorrer durante a execução do projeto. Esta etapa é composta pela atividade:

• **Revisões no Processo:** consiste em identificar, registrar e monitorar (através da ferramenta *Redmine*) os possíveis problemas que venham a surgir durante a execução do projeto, a fim de manter o acompanhamento da resolução destes.

Conforme descrito anteriormente, a fase de Execução de Processo inicia-se após a conclusão da etapa de Planejamento no módulo de *Gerência do Processo*. Os responsáveis (recursos humanos) alocados para uma atividade no processo, ainda no módulo de *Gerência do Processo*, são considerados *Usuários do Processo*. No módulo

de *Execução do Processo*, os recursos humanos alocados poderão executar de forma semi-automatizada e flexível as atividades do processo. Este módulo foi desenvolvido a partir do uso da biblioteca *JGraph*, que permite representar graficamente o modelo do processo a ser executado. Este módulo é responsável por gerar os produtos de trabalho que se fazem necessários para a condução das atividades do projeto. O versionamento destes produtos de trabalho é gerenciado através do apoio do *Subversion*.

O módulo de *Execução do Processo* também consiste na Aplicação do Formalismo de Execução, onde a ferramenta Spider-PE é responsável pela aplicação do formalismo de execução que é definido na especificação técnica da xSPIDER_ML [Portela e Gomes, 2011b]. Esta fase deve ocorrer em paralelo às etapas de Execução e Monitoramento no módulo de *Gerência do Processo*, pois este formalismo é executado simultaneamente às atividades deste módulo através da máquina de execução. Atualmente, o módulo de *Execução do Processo* está em fase de desenvolvimento no laboratório do Projeto SPIDER. Os módulos de *Administração* e *Gerência do Processo* já foram desenvolvidos e testados.

4. Resultados Esperados

Espera-se, a partir da adoção da ferramenta Spider-PE na execução de processos, que as organizações desenvolvedoras de software obtenham os seguintes resultados:

- Execução de modelos de processos baseados nas notações do padrão SPEM no que tange a modelagem de processos de software, a partir da sua customização pela especificação da linguagem SPIDER_ML;
- Flexibilidade na execução do processo a partir da sistematização do formalismo da linguagem xSPIDER_ML. Este mesmo formalismo permite com que a execução ocorra de forma semi-automatizada;
- Aderência às recomendações dos modelos de qualidade CMMI-DEV e MR-MPS
 no que diz respeito às práticas de execução do processo, como foco em atender
 níveis de Capacidade do Processo. Por conseguinte, espera-se estimular que um
 maior número de organizações desenvolvedoras de software possam
 implementar e obter avaliação oficial destes modelos;
- Facilidade no acompanhamento da execução do processo de software pelos gerentes e usuários do processo;
- Redução de custos, durante a fase de execução, devido a proposta utilizar somente tecnologias livres, sem custos para a organização na aquisição destes softwares;
- Fornecimento de uma base de dados para o armazenamento e consulta às informações e aos artefatos gerados durante a execução;
- Possibilidade de customização da ferramenta, por se tratar de um software livre, para atendimento das diferentes culturas organizacionais.

A partir do apoio ferramental da Spider-PE, espera-se também que o suporte ao registro e à análise dos resultados da execução do processo seja sistematizado e que a implementação das atividades do *framework*, descrito na Subseção 2.3, incorpore boas práticas dos modelos de qualidade na execução do processo.

5. Considerações Finais

A ferramenta de software livre Spider-PE tem o intuito de apoiar a execução flexível e semi-automatizada de processos de software a partir das boas práticas definidas pelos modelos de qualidade e melhoria do processo. Desta maneira, pretende-se facilitar a adoção destes modelos de qualidade pelas organizações desenvolvedoras de software através da utilização desta ferramenta. Ressalta-se, como ponto forte desta proposta, o fato da ferramenta ser *opensource*, viabilizando que a comunidade acadêmica e/ou indústria possa contribuir para a evolução e melhoria da mesma.

O uso da ferramenta também objetiva auxiliar a indústria de software a atingir níveis mais satisfatórios de disciplina a partir da combinação de técnicas e métodos que auxiliam na execução do processo de maneira a automatizar a realização das atividades do projeto e disponibilizar as informações sobre o andamento do projeto.

O desenvolvimento da ferramenta Spider-PE está em andamento, mas como resultados preliminares tem-se: o *framework* de execução de processo [Portela e Gomes, 2011a]; o formalismo da linguagem xSPIDER_ML [Portela e Gomes, 2011b]; e os módulos de *Administração* e *Gerência do Processo* da ferramenta Spider-PE. Como um dos trabalhos futuros, encontra-se planejada a utilização da ferramenta em uma organização parceira do Projeto SPIDER, com foco na implementação do programa de melhoria da qualidade do processo organizacional. A partir disso, será concebido um relato de experiência do uso da ferramenta em um cenário organizacional.

6. Agradecimentos

Este trabalho está recebendo o apoio financeiro do PIBIC CNPq/UFPA através do Edital 03/2011 – PROPESP – PIBIC - CNPq/FAPESPA/UFPA/AF/INTERIOR e do CNPq através de bolsa DTI do Edital MCT/CNPq/FNDCT n° 19/2009.

Referências

- Barros, R e Oliveira, S. (2010a) "SPIDER_ML: Uma Linguagem de Modelagem de Processos de Softwares", Em: Anais da II Escola Regional de Informática.
- Barros, R e Oliveira, S. (2010b) "Spider-PE: Uma Ferramenta de Apoio à Modelagem de Processos de Software", Em: Anais do VIII Encontro Anual de Computação.
- Bendraou, R. *et al.* (2007) "Definition of an Executable SPEM 2.0", Em: 14th Asia-Pacific Software Engineering Conference, páginas 390–397.
- Conradi, R. e Liu, C. (1995) "Process Modeling Languages: One or Many?", Em: Proceedings of the 4th European Workshop on Software Process Technology, páginas 98-118.
- Feiler, H. e Humphrey, S. (1993) "Software Process Development and Enactment: Concepts and Definitions", Em: International Conference on the Software Processes IEEE Computer Society Press.
- França, B. *et al.* (2009) "Utilização do Ambiente *WebAPSEE* na implantação do nível G do MPS.BR no CTIC-UFPA", Em: Anais do VIII Simpósio Brasileiro de Qualidade de Software
- GNU Project (2012) "General Public License", http://www.gnu.org, Março.
- Guizzardi, G., Falbo, R. e Guizzardi, R. (2008) "Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The Case of the ODE

- Software Process Ontology", Em: Anais do XI Iberoamerican Workshop on Requirements Engineering and Software Environments.
- Humphrey, S. (1989) "Managing the Software Process", The SEI Series in Software Engineering. Addison Wesley.
- Lima, A. *et al.* (2006) "WebAPSEE: Um Ambiente Livre e Flexível Para Gerência de Processos de Software", Em: Anais do VII Workshop de Software Livre.
- Lomchamp, J. (1993) "A Structured Conceptual and Terminological Framework for Software Process Engineering.", Em: The Second International Conference on the Software Process: Continuous Software Process Improvement, páginas 41-53.
- Oliveira, S. (2007) "ProDefiner: Uma Abordagem Progressiva para a Definição de Processos de Software no Contexto de um Ambiente Centrado no Processo", Tese de Doutorado em Ciência da Computação Centro de Informática, Universidade Federal de Pernambuco.
- Oliveira, S., Vasconcelos, A. e Rouiller, A. (2005) "Uma Proposta de Ambiente de um Implementação de Processo de Software", Em: Infocomp Journal of Computer Science, Vol. 4, páginas 71 78.
- Oliveira, S. *et al.* (2011) "Uma Proposta de Solução Sistêmica de um SUITE de Ferramentas de Software Livre de Apoio à Implementação do Modelo MPS.BR", Em: Revista do Programa Brasileiro de Qualidade e Produtividade em Software, páginas 103-107.
- Portela, C. e Gomes, M. (2011a) "Spider-PE: Definição do *Framework* de Execução de Processos", http://www.spider.ufpa.br/projetos/spider_pe/SPIDER-PE_Framework_Execucao_Processos.pdf, Março.
- Portela, C. e Gomes, M. (2011b) "xSPIDER_ML: Especificação Técnica.", http://www.spider.ufpa.br/projetos/xspider_ml/xSPIDER_ML.pdf, Março.
- Portela, C. *et al.* (2012) "xSPIDER_ML: Proposal of a Software Processes Enactment Language Compliant with SPEM 2.0", Em: Journal of Software Engineering and Applications, No prelo.
- Pressman, S. (2006) "Engenharia de Software", McGraw-Hill, 15ª Edição.
- Pressman, S. (2010) "Software Engineering: A Practioner's Approach", McGraw-Hill. 7ª Edição.
- Reis, C. (2000) "Ambientes de Desenvolvimento de Software e seus Mecanismos de Execução de Processos de Software", Exame de Qualificação do Doutorado Instituto de Informática, Universidade Federal do Rio Grande do Sul.
- Reis, C. (2003) "Uma Abordagem Flexível para Execução de Processos de Software Evolutivos", Tese de Doutorado em Ciência da Computação Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- SEI Software Engineering Institute (2010) "CMMI for Development V 1.3. 2010", http://www.sei.cmu.edu/reports/10tr033.pdf, Abril.
- SOFTEX Associação para Promoção da Excelência do Software Brasileiro (2011) "MPS.BR: Guia Geral 2011", http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_Geral_2011.pdf, Abril.
- Travassos, G. (1994) "O Modelo de Integração de Ferramentas da Estação TABA", Tese de Doutorado COPPE/Universidade Federal do Rio de Janeiro.
- The Standish Group International (2009) "Chaos Summary 2009".