

Modelo Comportamental de um Ambiente de Implementação de Processo de Software

SANDRO RONALDO BEZERRA OLIVEIRA ^{1,2}
ALEXANDRE MARCOS LINS DE VASCONCELOS ¹

¹Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851 – 50732-970 – Recife – PE – Brasil
Fone / Fax: (+55 81) 2126-8430

²Centro de Ciências Exatas e Tecnologia – Universidade da Amazônia (UNAMA)
Av. Alcindo Cacela, 287, 66060-902 – Belém – PA – Brasil
Fone: (+55 91) 4009-3000

(srbo, amlv)@cin.ufpe.br

Resumo. Ambiente de Engenharia de Software Centrado no Processo (PSEE - Process-centered Software Engineering Environment) tem como um dos seus propósitos prover que as fases do ciclo de vida de um processo de software (definição, simulação, execução e avaliação) possam ser automatizadas. Este trabalho apresenta um fluxo de serviços para a automação e implementação de processos de software, usando como base a especificação inicial de um ambiente [18]. Este processo deve ser instanciado a partir de características e propriedades que definem uma organização ou um domínio específico de projeto de software.

Palavras-Chave: Processo de Software, Automação de Processo de Software, Qualidade de Processo de Software, Ambientes de Desenvolvimento de Software, Fluxo de Atividades.

Dynamic Workflow of a Software Process Implementation Environment

Abstract. PSEE - Process-centered Software Engineering Environment - PSEE has one of its intentions to provide that phases of the software process life cycle (definition, simulation, enacting and evaluation) can be automatized. This paper presents a workflow for the automation and implementation of software processes, using the initial specification of a environment [18]. This process must be instantiated from characteristics and properties that define an organization or a specific domain of software project.

Keywords: Software Process, Software Process Automation, Software Process Quality, Software Development Environments, Workflow.

(Received November 09, 2005 / Accepted January 20, 2006)

1. Introdução

O surgimento da tecnologia CASE (*Computer Aided Software Engineering*) - Engenharia de Software Auxiliada por Computador, exerceu um enorme impacto sobre a área. A idéia de utilizar software para

auxiliar a produção de software foi bem recebida pelos desenvolvedores. As ferramentas CASE proporcionam uma sólida estrutura às metodologias e métodos de desenvolvimento de software. Os ambientes integrados de desenvolvimento de software, ou simplesmente ambientes de

desenvolvimento de software (ADSs) representam uma evolução do conceito de CASE, definindo mecanismos de integração entre as ferramentas, evoluindo para apoiar todas as etapas do ciclo de vida.

O apoio à gerência do processo de desenvolvimento de software corresponde a um dos requisitos que são exigidos nos ADSs atuais. Técnicas e ferramentas são integradas para permitir o controle de cada uma das etapas que são necessárias para desenvolver um produto de software. Assim sendo, os ambientes de desenvolvimento de software orientados ao processo [21] surgiram com o objetivo de proporcionar uma estrutura computacional que gerencie o intercâmbio de informações entre os desenvolvedores, mesmo que estes estejam em localidades geograficamente dispersas, controlando as atividades realizadas, envolvendo assim os recursos consumidos, os prazos determinados, e as datas de início e término de cada atividade. A gerência de processos de software começou a ser tratada como um dos mais eficazes mecanismos para se obter um aumento na qualidade do software produzido.

Muito já se discutiu sobre as propriedades deste tipo de tecnologia, no entanto, percebe-se ao longo da execução do processo a partir destes ambientes de desenvolvimento que sua implementação nem sempre perfaz a realidade das características da organização ou do projeto desenvolvido por esta. Isto se deve ao fato de que os responsáveis pela definição do processo não dispõem de um guia completo contendo as suas reais necessidades de execução e estes, por si, indiquem as melhores práticas a serem instanciadas a partir de um processo padrão.

Neste contexto, é possível encontrar na literatura e na comunidade especializada inúmeros ambientes de desenvolvimento de software centrados no processo, onde nem todas as atividades que perfazem o ciclo de vida de um processo de software (definição, simulação, execução e avaliação) são atendidas em sua completude, como PROSOFT [22], Odyssey Share [24], Provision [20], Adele-Tempo [5], SPADE [3], ProcessWeaver [8], EPOS [17], Estação TABA [23].

No entanto o que se detectou, a partir da análise das características de cada um desses ambientes, é que nenhum desses se preocupa com alguns pontos-chave de discussão atual: definição de processo baseada na detecção de aspectos que caracterizem uma classe de projeto de software específico e propriedades que definem a estrutura de uma

organização; uso de lições aprendidas ao longo de definições de processos, possibilitando desta forma que o ambiente sugira ao usuário atributos de composição do processo; adequação de atividades que especifiquem melhoria contínua no processo de software; reutilização de processos de acordo com o nível de sua definição (a ser discutido nas seções a seguir); plena automação das atividades que definem um ciclo de vida de processo de software, seja usando ferramentas já existentes na comunidade, seja pela adaptação destas.

Neste contexto foi proposto a definição de um novo ambiente para a implementação de processo de software [18]. Este ambiente tende a possibilitar a especificação dos processos de acordo com o domínio do projeto específico e das características da organização; a instanciação do processo de software para propriedades dos projetos; sua simulação a partir dos parâmetros de configuração (prazo, pressões, custo, recursos, etc.); uma execução (automação) mais próxima do que se espera para um processo organizacional; e uma avaliação a partir da coleta de métricas desta execução. O projeto deste ambiente de implementação de processos de software e seu modelo comportamental serão descritos neste artigo.

Além desta seção introdutória, o artigo apresenta outras quatro seções. A seção 2 aborda as propriedades que compõem um ambiente de implementação de processo de software. Na seção 3 tem-se o detalhamento dos fluxos de atividades que formam os serviços providos pelo ambiente. A seção 4 aborda a necessidade do estudo de caso para os serviços providos pelo ambiente e relata um guia para a sua execução. Finalmente, a seção 5 apresenta as considerações finais deste artigo.

2. ImPProS – Ambiente de Implementação Progressiva de Processo de Software

Para ajudar uma organização na implementação progressiva de um processo de software, é útil fornecer apoio automatizado por meio de um ambiente capaz de suportar as fases que a literatura especializada propõe como necessárias. O termo “progressiva” decorre do fato de que a implementação do processo é aperfeiçoada com as experiências aprendidas na sua definição, simulação, execução e avaliação.

O ambiente ImPProS está sendo concebido com o objetivo principal de apoiar a implementação de um processo de software em uma organização. Dentro deste contexto podem ser caracterizados como seus objetivos específicos [18]:

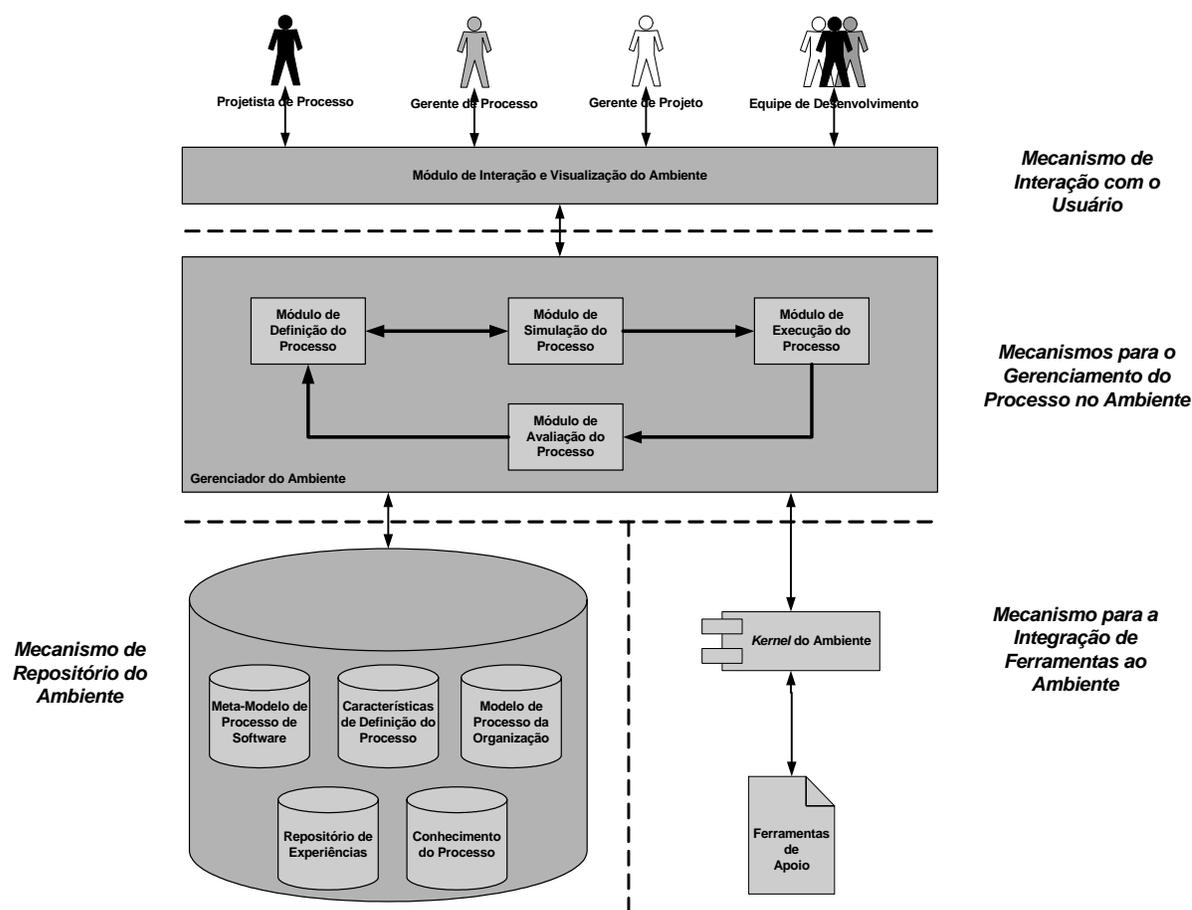


Figura 1. Arquitetura do Ambiente de Implementação de Processo de Software

- Especificar um meta-modelo de processo de software a fim de definir uma terminologia única entre os vários modelos de qualidade de processo de software existentes, para uso do ambiente em seus serviços providos.
- Apoiar a definição de um processo de software para organização;
- Permitir a modelagem e instanciação deste processo;
- Permitir a simulação do processo a partir das características instanciadas para um projeto específico;
- Dar apoio à execução do processo de software tomando como base uma máquina de inferência;
- Possibilitar a avaliação dos critérios do processo de software;
- Apoiar a melhoria contínua do processo de software e o reuso através da realimentação e coleta das experiências aprendidas.

Vale ressaltar que todos os objetivos listados acima foram adaptados a partir da estrutura que compõe o meta-processo de software descrito em [21], das características propostas para a implementação de um processo de software [2] e do ciclo de vida para melhoria contínua de processo definido pelo Modelo IDEAL [15]. Para alcançar estes objetivos o ambiente foi concebido para adotar a arquitetura apresentada na Figura 1. Os componentes definidos na arquitetura do ambiente encontram-se resumidamente descritos nas subseções a seguir.

2.1. Mecanismo de Interação com o Usuário

Neste mecanismo o foco está em prover aos usuários diferentes visões da mesma informação sendo definida e especificada, provendo interação para

diferentes usuários do ambiente, ou seja, trabalha com as características da usabilidade no ambiente.

2.2. Mecanismo para o Gerenciamento do Processo no Ambiente

Este mecanismo possui a responsabilidade de prover os serviços (módulos de definição, simulação, execução e avaliação do processo de software) especificados ao ambiente de forma automatizada, ou seja, possibilitar que os usuários do ambiente executem suas funções tendo como referencial um guia. Um detalhamento maior do funcionamento de cada um dos serviços providos por este mecanismo será abordado na seção 3.

O relacionamento dos serviços providos por cada um dos módulos de gerenciamento do ambiente estão representados na Figura 2 na forma de uma pirâmide que define, numa versão bottom-up, como se dá a execução dessas funcionalidades. Pode-se notar, ainda, que na coluna vertical representada na mesma figura estão definidas algumas ferramentas e/ou aplicativos que darão o suporte para que os serviços do ambiente sejam executados.

2.3. Mecanismo de Repositório do Ambiente

O foco deste mecanismo está em prover ao ambiente o sistema de gerenciamento dos seus objetos a partir de bases de dados que provejam o controle de evolução e manutenção dos componentes do processo de software.

2.4. Mecanismo para Integração de Ferramentas ao Ambiente

Este mecanismo provê a integração do ambiente com outras ferramentas de apoio ao processo de software e à execução do projeto de software, possibilitando a automação de atividades definidas no processo e a execução de alguns módulos do ambiente.

3. Fluxos de Trabalho do ImPProS

Os fluxos de trabalho (workflows) representam uma seqüência de atividades que são executadas em um negócio para produzir um resultado de valor para algum ator do negócio. Os fluxos de trabalho podem ser descritos informalmente ou formalmente. Fluxos de trabalho formais são descritos através de linguagens apropriadas, as WDLs (Workflow Description Languages). Algumas destas linguagens são derivadas de modelos conceituais conhecidos na engenharia de software como statecharts [14]. Outras, todavia, usam suas próprias estruturas conceituais como em [6]. Nós optamos por descrever os fluxos utilizando as primitivas descritas na Figura 3, por ser mais simples e permitir que sejam declarados os executores de cada atividade, e os limites de responsabilidade destes atores na execução das atividades.

As subseções a seguir detalham os fluxos de trabalhos que ilustram o relacionamento entre os módulos do ImPProS, detectados na Figura 1.

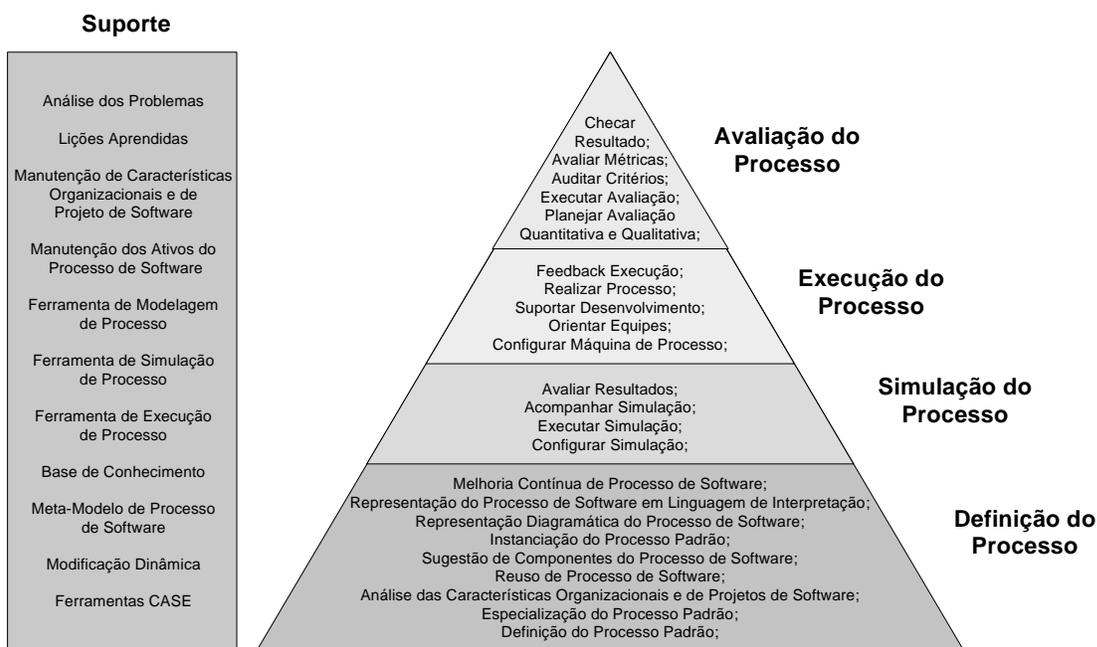


Figura 2. Relacionamento dos módulos de gerenciamento do ImPProS

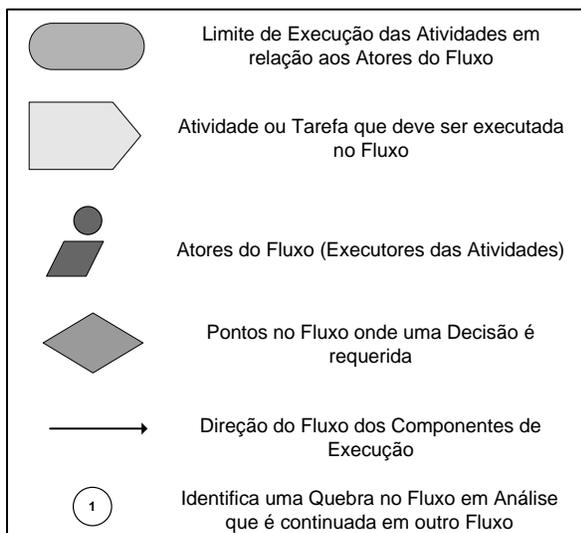


Figura 3. Primitivas do Fluxo de Trabalho

3.1. Fluxo de Trabalhos do Módulo de Definição do Processo

O módulo de Definição do Processo do Ambiente é responsável por prover a definição, modelagem e instanciação do processo de software para uma organização. O fluxo de trabalho deste módulo pode ser visualizado segundo a Figura 4. Podemos detectar uma descrição resumida dos serviços providos por cada atividade a seguir:

a) Manter Ativos do Meta-Modelo do Processo: o foco desta atividade é possibilitar que o Projetista do Processo faça a manutenção, sempre que necessária, dos componentes do meta-modelo de processo de software [16] definido no ambiente, ou seja, adaptar o meta-modelo, responsável por padronizar (mapear) as terminologias relacionadas ao processo de software tomando como base normas/modelos de qualidade de processo (por exemplo, CMMI [7], SPICE – ISO/IEC 15504 [12], ISO/IEC 12207 [13]). É importante ressaltar que cabe ao Projetista do Processo adequar o meta-modelo do processo de software levando em consideração a semântica dos componentes (descrição dos componentes) adaptados para que estes não se encontrem repetidamente representados no meta-modelo.

b) Manter Regras de Projetos de Software e da Organização: esta atividade é responsável por permitir que o Projetista do Processo efetue os inputs das características que definem um projeto de software [4] (equipe, paradigma de desenvolvimento, tipo de software, produto, complexidade, etc.) e uma organização [10] (ambiente de desenvolvimento,

maturidade, etc.). Estas características devem ser compostas de regras a fim de possibilitar a especificação dos vários níveis de processo a partir de sugestões feitas pelo ambiente.

c) Definir Processo Padrão para a Organização: atividade que possibilita adaptar a uma organização os componentes do meta-modelo de processo de software de acordo com as suas características de desenvolvimento, maturidade e capacitação do processo de software. O ambiente faz sugestões de componentes específicos do processo (quando não há processo a ser reusado pelo ambiente) ou de processo completo (quando há processos para reuso que se assemelham pelas características especificadas) para esta adaptação, e o Projetista do Processo pode acatar ou não. Caso haja necessidade de uma adaptação, o ambiente possibilita um armazenamento das experiências aprendidas a fim de prover a melhoria contínua no processo de software.

d) Definir a Especialização do Processo Padrão: nesta atividade, o Projetista do Processo, com base nas características do software e do paradigma usado, especializa o processo padrão especificado para atender as necessidades específicas de cada paradigma ou tipo de software a ser usado como base para o desenvolvimento, ou seja, determinar quais os componentes do processo devem ser adequados ao paradigma e com as características do software definidos. Semelhantemente à atividade anterior, o ambiente faz sugestões de componentes específicos do processo.

e) Definir a Instanciação do Processo Padrão: esta atividade possibilita que o Projetista do Processo instancie o processo especializado na atividade anterior para atender as características do projeto (equipe, qualidade do produto, ciclo de vida, métodos, tipo de ferramentas, recursos, etc.) a ser executado, ou seja, a instanciação provê apenas a definição dos componentes (tipos de ferramentas, ciclo de vida, métodos, etc.) necessários que formarão o processo e não os componentes de projeto (prazos, ferramentas, pessoas, etc.). Assim, como as duas atividades anteriores, o ambiente faz sugestões de componentes específicos do processo.

f) Gerar Modelagem do Processo: esta atividade provê a representação diagramática do processo de software instanciado para um projeto específico a partir da notação usada pela linguagem de modelagem de processo padrão no ambiente [1]. O ambiente faz uso de uma ferramenta de modelagem de processo para a automação desta atividade. O ambiente, também, provê ao Projetista do Processo a

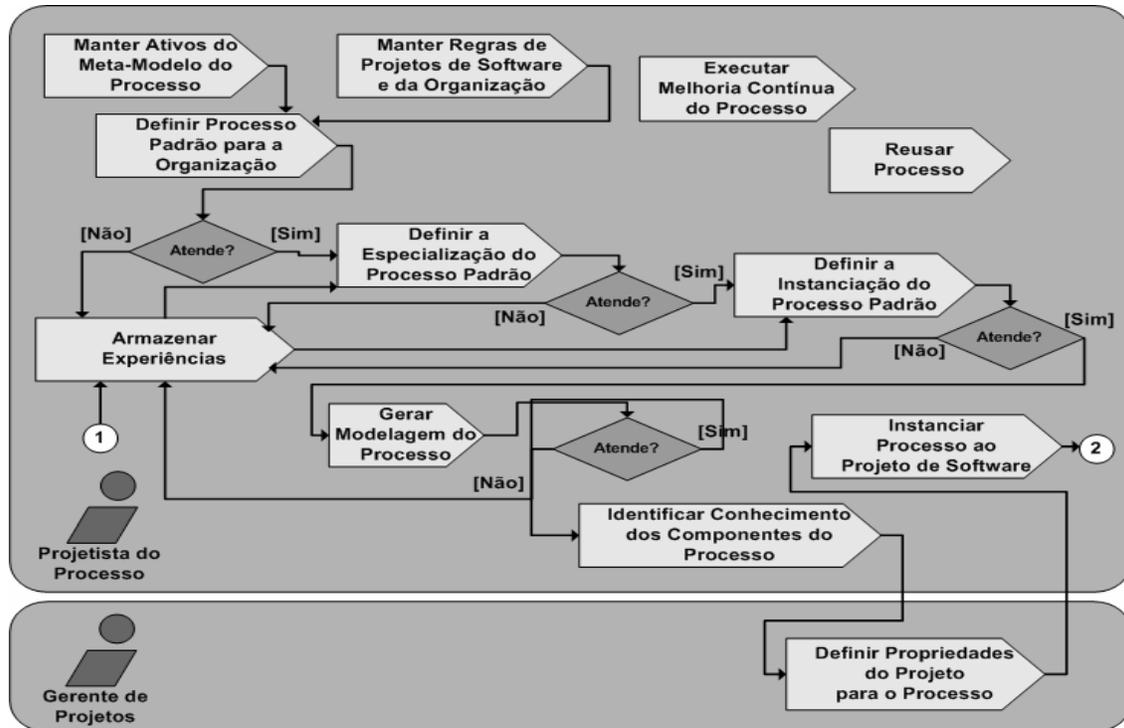


Figura 4. Fluxo de Trabalhos do Módulo de Definição do Processo

análise da representação efetuada a fim de possibilitar ajustes e fazendo com que o ambiente aprenda com as experiências armazenadas do processo de software.

g) *Armazenar Experiências*: esta atividade permite que o ambiente mantenha experiências formais e informais aprendidas na definição, modelagem, simulação e avaliação da execução do processo de software para (semi-)automatizar a execução das atividades de adaptação, especialização e instanciação de novos processos e possibilitar a melhoria contínua de processos existentes, a partir das sugestões inferidas pelo ambiente acerca do definido pelos usuários.

h) *Identificar Conhecimento dos Componentes do Processo*: esta atividade permite que o ambiente identifique as notações que perfazem a representação diagramática do processo de software, a partir da ferramenta de modelagem do processo, a fim de compor as regras de conhecimento¹ para a execução do processo. A finalidade desta atividade está no fato

de que as ferramentas de modelagem do processo não geram, a partir da notação, a lógica necessária para a execução automatizada do processo a partir de uma máquina de processos.

i) *Definir Propriedades do Projeto para o Processo*: define um plano de execução do processo de software a partir dos componentes identificados na representação diagramática, ou seja, aloca aos componentes do processo as propriedades de execução do projeto (prazos, ferramentas, pessoas, etc.) e como estes deverão ser executados (plano de iterações se for necessário, etc.).

j) *Instanciar Processo ao Projeto de Software*: esta atividade possui como um dos focos o desenvolvimento de um plano de execução do processo a um projeto específico de acordo com os componentes especificados para a sua composição, ou seja, é gerado um modelo de processo instanciado, contendo informações detalhadas sobre os prazos, agentes e recursos utilizados por cada atividade definida no processo. Além disso, com base nas regras de conhecimento identificadas e no plano de execução do processo definido, o ambiente instancia o processo ao projeto de software, a ser desenvolvido, a partir do conjunto de ontologias previamente composta [11].

l) *Executar Melhoria Contínua do Processo*: esta atividade provê o aperfeiçoamento contínuo do

¹ Regras de Conhecimento são as definições em linguagem natural dos componentes diagramáticos do modelo de processo de software e servem para serem mapeadas para a linguagem que servirá como base de execução do processo de software por uma máquina de processo.

processo de software através da análise do fluxo de trabalhos do modelo IDEAL. Algumas destas funções são [9]: definição do estímulo de melhoria e contexto do aperfeiçoamento; caracterização do estado atual e futuro; verificação das prioridades de melhoria; desenvolvimento da estratégia; definição do plano de implementação; criação, refinamento e implementação da solução; análise e validação da melhoria; e propostas de ações futuras. É executada, sempre que necessário, paralelamente a todas às outras atividades.

m) Reusar Processo: o propósito desta atividade está em prover o reuso de modelos de processo de software durante sua definição. Este procedimento ocorre a partir da análise de características (de projetos de software e organizacionais) específicas para a definição, especialização e instanciação do processo padrão. Suas funções são [19]: análise das características do processo; identificação do nível de processo a ser reusado; identificação do processo a ser reusado; propor o reuso de processo; análise da sugestão de reuso; recuperar processo. É executada, sempre que necessário, paralelamente a todas às outras atividades.

3.2. Fluxo de Trabalhos do Módulo de Simulação do Processo

Este módulo é responsável por antever os resultados da execução do processo. Na Figura 5 pode-se visualizar o fluxo de trabalhos que compõe a

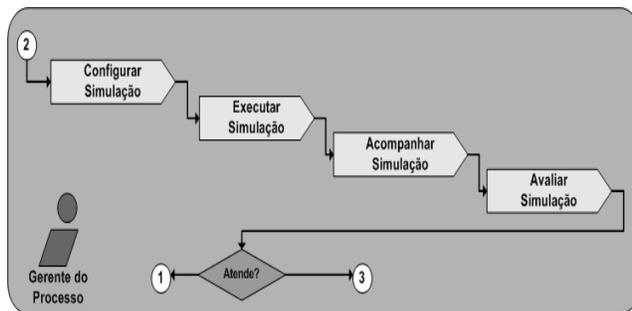


Figura 5. Fluxo de Trabalhos do Módulo de Simulação do Processo

execução dos serviços deste módulo. Podemos detectar uma descrição resumida dos serviços providos a seguir:

a) Configurar Simulação: esta atividade faz com que o processo especificado, a partir das regras de conhecimento, seja configurado na ferramenta de apoio à simulação atendendo todas as definições especificadas no plano de execução do processo.

b) Executar Simulação: esta atividade permite a execução da simulação do processo de software de acordo com o especificado no plano de execução do processo e configurado na ferramenta de simulação integrada ao ambiente.

c) Acompanhar Simulação: esta atividade possibilita que o Gerente de Processo monitore a simulação do processo de software de acordo com os termos especificados no plano de execução do processo. Esta atividade deve possibilitar que o Gerente de Processo faça uso de algum mecanismo a fim de registrar os pontos relevantes desta simulação para futuras análises.

d) Avaliar Simulação: paralelamente à execução da simulação do processo, a ferramenta de simulação provê ao Gerente de Processo os resultados obtidos com esta simulação para sua avaliação. Caso o resultado obtido não seja o esperado, o Gerente de Processo solicita o armazenamento, no ambiente, das experiências aprendidas com esta simulação, a fim de possibilitar a melhoria contínua do processo. Caso contrário, a ferramenta de simulação armazena o processo simulado para sua execução por uma máquina de processo.

3.3. Fluxo de Trabalhos do Módulo de Execução do Processo

O módulo de Execução do Processo do Ambiente é responsável pela automação, visualização e representação do processo de software a partir das características de um projeto específico. O fluxo de trabalho deste módulo pode ser visualizado na Figura 6. Podemos detectar uma descrição resumida dos serviços providos a seguir:

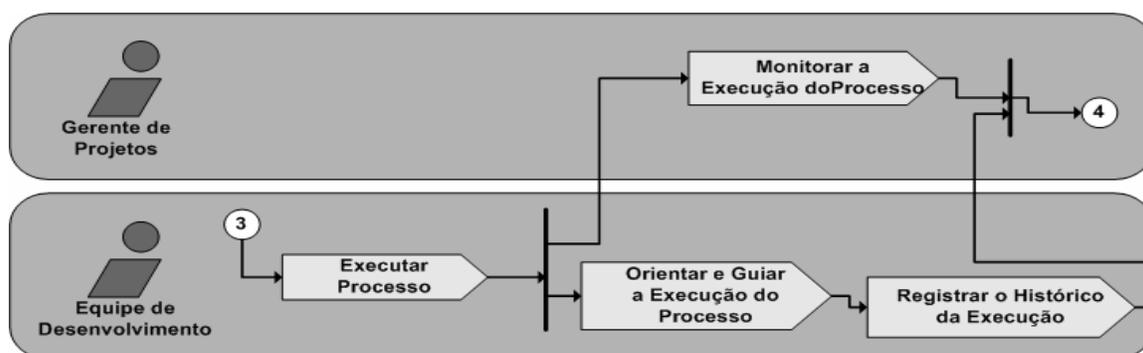


Figura 6. Fluxo de Trabalhos do Módulo de Execução do Processo

a) *Executar Processo*: nesta atividade, o ambiente, de posse de uma máquina de processo, deve alimentá-la com o processo simulado a fim de interpretar as regras compostas por conhecimentos do processo e assim prover a execução do processo de forma automatizada. Esta atividade possui como responsabilidade a ativação de ferramentas CASE para possibilitar a execução do projeto a partir das especificações no processo.

b) *Orientar e Guiar a Execução do Processo*: esta atividade funciona como um guia para a equipe de desenvolvimento. Através da representação e visualização do processo de software, esta atividade orienta a equipe de desenvolvimento na execução de suas tarefas do projeto em desenvolvimento.

c) *Registrar o Histórico da Execução*: esta atividade funciona como uma agenda capaz de fornecer à equipe de desenvolvimento mudança no status de execução dos componentes que formam o processo de software, guiando a seqüência de ações definidas. Possibilita o registro de todas as ações tomadas na execução do processo.

d) *Monitorar a Execução do Processo*: o foco está em possibilitar ao Gerente de Projetos um acompanhamento da execução do projeto a partir das características definidas no plano de processo e o status da execução dos componentes que formam o processo de software em uso.

3.4. Fluxo de Trabalhos do Módulo de Avaliação do Processo

Este módulo é responsável por avaliar qualitativamente e quantitativamente a execução do processo possibilitando uma análise quanto a necessidade de novos requisitos ao processo. Na Figura 7 pode-se visualizar o fluxo de trabalhos que compõe a execução dos serviços deste módulo. A seguir tem-se uma descrição resumida dos serviços:

a) *Coletar Métricas*: esta atividade provê ao ambiente a coleta de métricas a partir dos registros efetuados ao longo da execução do processo de software, a fim de prover ao Gerente de Processo futuras tomadas de decisão (avaliações qualitativas e quantitativas) em relação ao processo em execução.

b) *Avaliar a Execução do Processo*: com as métricas coletadas na atividade anterior, o Gerente de Processo avalia qualitativamente e quantitativamente o andamento da execução do processo auxiliando-o na tomada de decisões.

c) *Avaliar a Necessidade de Modificação Dinâmica*: uma das tomadas de decisão do Gerente de Processo recai na modificação dinâmica do processo a partir do armazenamento de experiências aprendidas para prover a melhoria contínua do mesmo. Se houver esta necessidade, novos requisitos do processo de software devem ser passados para o módulo de Definição do Processo para análise, coleta das lições aprendidas e avaliação da necessidade ser contemplada no processo.

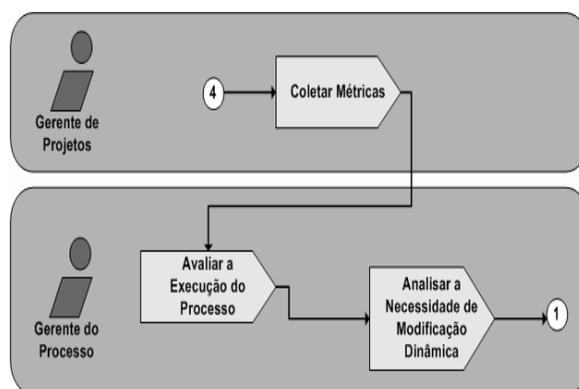


Figura 7. Fluxo de Trabalhos do Módulo de Avaliação do Processo

4. Relato da Necessidade do Estudo de Caso

Embora a estrutura apresentada do ambiente possa estar caracterizada em um nível de completude capaz de atender as necessidades de todas as atividades do ciclo de vida de um processo de software [22], nem todos os módulos de serviços providos foram validados a partir de um estudo de caso real. Sendo apenas analisada sua importância e viabilidade a partir das características que compõem os ambientes citados na seção 1.

No entanto, planeja-se, como ação futura, uma experimentação geral das idéias propostas neste ambiente com base em um roteiro definido como eficaz pelo grupo de pesquisa, aliado ao uso de algumas empresas regionais que tenham como foco o desenvolvimento de produtos de software.

Este roteiro consiste dos seguintes passos: análise das características organizacionais e dos tipos de projetos de software desenvolvidos; definição do processo de software com base em sugestões do ambiente e seguindo os três níveis de definição do processo de software (processo padrão, processo especializado e processo instanciado); modelagem diagramática deste processo para uma melhor visualização dos seus componentes; instanciamento do mesmo usando-se propriedades (tempo, pessoas, custo, etc.) de um projeto de software; captura das lições aprendidas a fim de propiciar a melhoria contínua e a reutilização de processos definidos; simulação do processo usando-se uma ferramenta de suporte; execução do processo a partir de uma máquina de inferência; avaliação desta execução para uma análise do que fora implementado.

Espera-se como resultado desta validação um atendimento global de todas as funções que compõem os módulos do ambiente de implementação de processo de software (ver Figura 1) e sua melhor adequação às necessidades reais de uma organização de desenvolvimento de produtos de software.

Vale ressaltar que alguns módulos já possuem validações, a partir de estudos de casos reais, individuais dos serviços providos em sua concepção, a saber: ferramenta de coleta de experiências; ferramenta de análise de problemas dos itens do processo de software; ferramenta de reuso de processo de software; ferramenta de melhoria contínua do processo de software; ferramenta de análise das características organizacionais e de projetos de software para sugestão de componentes do processo de software; ferramenta de composição e

mapeamento do meta-modelo de processo de software).

5. Considerações Finais

Como qualquer produto resultante de uma atividade de engenharia, é esperado que os produtos de software tenham características intrínsecas de qualidade perceptíveis pelo usuário. A qualidade de software pode ser vista como um conjunto de características que devem ser alcançadas em um determinado grau para que o produto atenda às necessidades de seus usuários.

No entanto, observou-se que a qualidade do produto de software está ligada ao processo de geração do software. Em outras palavras, não é possível adicionar qualidade a um software depois de pronto. Assim, na década de 90, houve uma grande preocupação quanto à qualidade do processo das organizações de desenvolvimento de software. Essa preocupação resultou em modelos para avaliar e melhorar os processos de software, cujo objetivo é dar uma indicação da maturidade de um processo de software e definir ações para evolui-lo.

Espera-se que ao longo dos anos as organizações de desenvolvimento de software ajustem seus processos de software para a produção de produtos de qualidade dentro de prazos confiáveis. Mais ainda, estas organizações serão pressionadas constantemente a otimizar os seus processos de desenvolvimento e manutenção, de forma a produzir produtos cada vez a custos menores e com qualidade crescente.

Assim, um ambiente capaz de prover a implementação progressiva de processos de software a partir da definição, simulação, execução e avaliação deste processo é de fundamental importância para que o cenário de aperfeiçoamento dos processos das organizações de desenvolvimento de software seja mais amenizado com a automação das atividades, representadas na forma de fluxos, listadas neste trabalho.

Atualmente o ambiente encontra-se em fase de desenvolvimento de alguns dos seus módulos (definição, simulação, avaliação e modelagem de processo de software), definidos na Figura 1, e outros (reuso, melhoria contínua, meta-modelo de processo de software, coleta de experiências, análise de problemas, análise de características organizacionais e de projetos de software) já apresentam validações de seus usos a partir de projetos reais. São usados para composição (representação), de algumas atividades do ambiente, tecnologias consideradas *free*

e *open source*, as quais são integradas ao ambiente a partir de um plugin de comunicação, o que pode-se destacar como mais um diferencial do ambiente em relação aos existentes na literatura especializada, alavancando o uso nas empresas de pequeno e médio porte do processo de software.

Referências Bibliográficas

- [1] Araujo, R. M. Construção Gráfica de Processos de Desenvolvimento e Geração de uma Ontologia de Processo de Software, Trabalho de Graduação apresentado ao CIn/UFPE, orientador Prof. Alexandre Vasconcelos, Recife-PE, 2005.
- [2] Balduino, R. Implementação de um processo de desenvolvimento de software: uma abordagem passo-a-passo, Rational Software White Paper, 2002.
- [3] Bandinelli, S. et al. SPADE: An Environment for Software Process Analysis, Design and Enactment, In: FINKELSTEIN, A. et al. (Ed.). Software Process Modelling and Technology. Taunton: Research Studies Press, 1994.
- [4] Barbosa, I. M. Análise de Características de Projetos de Software para a Definição de Processo de Software, Trabalho de Graduação apresentado ao CIn/UFPE, orientador Prof. Alexandre Vasconcelos, Recife-PE, 2005.
- [5] Belkhatir, N., Estublier, J., Melo, W. ADELE-TEMPO: an Environment to Support Process Modelling and Enaction, In: FINKELSTEIN, A. et al. (Ed.). Software Process Modelling and Technology. Taunton: Research Studies Press, 1994.
- [6] Casati, F., Ceri, S., Pernici, B. and Pozzi, G. Conceptual Modelling of Workflows, International Conference on Object-Oriented and Entity-Relationship - OOEF'95, Gold Coast, Austrália, 1995.
- [7] Chrissis, M. B., Konrad, M. and Shrum, S., CMMI Guidelines for Process Integration and Product Improvement, Addison-Wesley, 2003.
- [8] Christie, A. Software Process Automation: The Technology and its adoption, Berlin: Springer Verlag, 1997.
- [9] Correia, R. S. M. Análise e Adaptação do Modelo IDEAL para um Ambiente de Implementação de Processo de Software, Trabalho de Graduação apresentado ao CIn/UFPE, orientador Prof. Alexandre Vasconcelos, Recife-PE, 2005.
- [10] Cunha, M. B. F. L. Análise das Características Organizacionais para a Definição de Processo de Software, Trabalho de Graduação apresentado ao CIn/UFPE, orientador Prof. Alexandre Vasconcelos, Recife-PE, 2005.
- [11] Falbo, R. A. Integração de Conhecimento em um Ambiente de Desenvolvimento de Software”, Orientadora: Ana Regina Cavalcanti da Rocha. Tese de Doutorado, COPPE/UFRJ, 1998.
- [12] ISO/IEC TR 15504, Parts 1-9 Information Technology – Software Process Assessment, International Organization for Standardization, 1998.
- [13] ISO/IEC TR 12207 Amendment: Information Technology – Amendment to ISO/IEC 12207, PDAM 3 version, 2000.
- [14] Kappel, G., Rusch-Schott, S. and Retschitzegger, W. Rule Patterns for Designing Active *Object-Oriented Database Applications*, Technical Report TR-07/95, Linz – Austria, 1995.
- [15] Mcfeeley, B. IDEALSM: A User’s Guide for Software Process Improvement, Software Engineering Institute Handbook. Carnegie Mellon University. CMU/SEI-96-HB-001, 1996.
- [16] Mendes, R. C. Modelagem e Avaliação do CMMI no SPEM para Definição de uma Meta-Modelo de Processo de Software, Trabalho de Graduação apresentado ao CIn/UFPE, orientador Prof. Alexandre Vasconcelos, Recife-PE, 2005.
- [17] Nguyen, M. N., Wang, A. I. Total Software Process Model in EPOS, disponível por www em: <http://www.idt.unit.no/~epos/Papers>, 1996.
- [18] Oliveira, S. R. B., Vasconcelos, A. M. L., Rouiller, A. C. Uma Proposta de um Ambiente de Implementação de Processo de Software, Revista InfoComp – Revista de Ciência da Computação da UFLA – vol. 4, n. 1, Lavras-MG, 2005.
- [19] Oliveira, E. H. R. Reuso de Processo de Software em um Ambiente de Implementação de Processo de Software, Trabalho de Graduação apresentado ao CIn/UFPE, orientador Prof. Alexandre Vasconcelos, Recife-PE, 2005.
- [20] Proforma Business Process Improvement, disponível por www em: <http://www.proformacorp.com/downloads/whitepapers.asp>, 2000.
- [21] Reis, C. A. L. Ambientes de Desenvolvimento de Software e seus Mecanismos de Execução de Processos de Software, Orientador Daltro Nunes.

Exame de Qualificação do Doutorado. PPGC–UFRGS, 2000.

- [22]Reis, C. A. L. Uma Abordagem Flexível para Execução de Processos de Software Evolutivos, Orientador Daltro Nunes. Tese de Doutorado. PPGC–UFRGS, 2003.
- [23]Rocha, A. R. C., Maldonado, J. C. and Weber, K. C., Qualidade de software: teoria e prática, São Paulo: Prentice-Hall, 2001.
- [24]Werner, C. M. L., Mangan, M. A. S., Murta, L. G. P., Souza, R. P., Mattoso, M., Braga, R. M. M., Borges, M. R. S. OdysseyShare: an Environment for Collaborative Component-Based Development, In: Information Reuse and Integration (IRI), Las Vegas, Nevada, USA, 2003.