

Workflow Aplicado a Engenharia de Software Baseada em Processos: Uma Visão Geral

RENATO AFONSO COTA SILVA
LIZIANE SANTOS SOARES
JOSÉ LUIS BRAGA

Departamento de Informática (DPI) – Universidade Federal de Viçosa (UFV)
Av. P.H. Rolfs s/n – Campus UFV – 36570-000 – Viçosa, MG
(renatoacs, liziane, zeluis)@dpi.ufv.br

Resumo. Com a crescente necessidade de aumentar a produtividade no desenvolvimento de software visando uma maior qualidade do produto final, faz-se necessário a adoção de novas técnicas de desenvolvimento. Utilizando-se técnicas de Ambientes de Desenvolvimento de Software Orientados a Processo (*Process-Centered Software Engineering Environments* - PSEE) temos uma organização mais sistemática do processo, facilitando sua gerência. A gerência de processo ajuda os projetos de software a alcançar disciplina, controle e entendimento claro dos seus processos e atividades. Para viabilizar a gerência de processos podemos aplicar tecnologia de *workflow* em Ambientes de Desenvolvimento de Software Orientados a Processo. A gerência de um *workflow* pode ser realizada pelos Sistemas de Gerenciamento de Workflow (*Workflow Management Systems* - WFMS), que oferecem ferramentas para modelar, executar e gerenciar *workflows*.

Palavras-Chave: Workflow, Engenharia de Software, Ambiente de Engenharia de Software

Workflow Applied the Software Engineering Process Based: A Survey

Abstract. The need to increase software development productivity and quality demands the adoption of new development techniques. Process-Centered Software Engineering Environments-PSEE is a promising one, providing a more systematic process organization and management. Good process management techniques helps software developers to reach higher levels of discipline and quality and improves the level of control over process activities and documentation. Workflow technologies are usually included as part of PSEEs, providing automatic (or semi-automatic) flow control, increasing process reliability by using Workflow management systems - WFMS, that provide tools to model, execute and manage workflows.

Keywords: Workflow, Software Engineering, Software Engineering Environment

(Received January 27, 2006 / Accepted May 9, 2006)

1 Introdução e motivação

Dentre as principais áreas que constituem a Ciência da Computação, uma das que mais influenciam o mundo atual é a Engenharia de Software, envolvida nos aspectos

tecnológicos e gerenciais do processo de desenvolvimento de software ou simplesmente do processo de software [7]. Software tornou-se a base de sustentação de inúmeras organizações dos mais diversos ramos de atuação espalhados pelo planeta, consistindo no ele-

mento estratégico da diferenciação de produtos e serviços. Atualmente, o software está embutido em sistemas de uma infindável lista de diferentes ciências e tecnologias, e será o propulsor dos novos avanços que influenciam uma ampla gama de indústrias, envolvendo desde a educação elementar até corporações envolvidas com a engenharia genética [15].

Apesar dos inúmeros avanços recentes nesta área, muito ainda é discutido acerca da baixa qualidade dos produtos e produtividade da indústria mundial de software, refletindo-se na insatisfação dos seus usuários e em prejuízos financeiros de enormes proporções. Por outro lado, os computadores já são componentes comuns do dia-a-dia das pessoas, o que faz com que as os requisitos dos usuários cresçam em ordem de complexidade.

Com o objetivo de solucionar esses problemas, uma das principais linhas de pesquisa que vem sendo desenvolvida para melhorar a qualidade dos softwares é o estudo e melhora dos processos de desenvolvimento e manutenção de softwares, onde várias tecnologias vêm sendo experimentadas [21] no sentido de apoiar o ciclo de vida do software. Um dos esforços mais significativos corresponde à definição de metodologias para disciplinar o processo de desenvolvimento pelo estabelecimento de etapas bem definidas, proporcionando um mecanismo de controle para o processo. É fato conhecido que existe uma relação direta entre qualidade do processo e qualidade do produto final que é o software. Como consequência, muitas organizações de desenvolvimento de software buscam a maturidade no processo de software, usando estruturas ou "frameworks" como o CMM - *Capability Maturity Model for Software* [14] para estruturar as iniciativas de melhoria de processo.

A Tecnologia de Processo de Software surgiu no final da década de 1980 e representou um importante passo em direção à melhoria da qualidade de software através de mecanismos que proporcionam o gerenciamento automatizado do desenvolvimento de software [5]. Diversas teorias, conceitos, formalismos, metodologias e ferramentas surgiram nesse contexto enfatizando a descrição de um modelo de processo de software que é automatizado por um ambiente integrado de desenvolvimento.

Informalmente, o processo de software pode ser compreendido como o conjunto de todas as atividades necessárias para transformar os requisitos do usuário em software [9, 13]. O processo de software é formado por um conjunto de passos parcialmente ordenados relacionados com conjuntos de artefatos, pessoas, recursos, estruturas organizacionais e restrições, tendo como objetivo produzir e manter os produtos de software re-

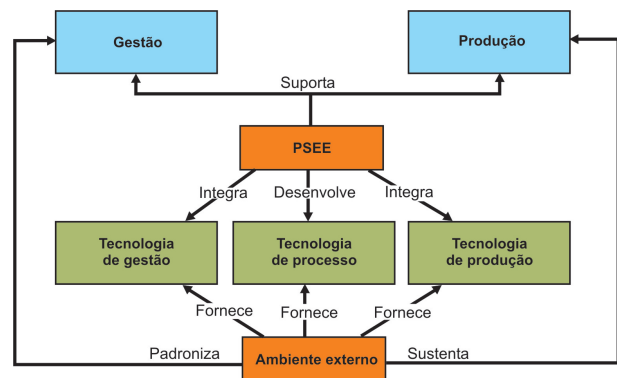


Figura 1: Tecnologia do Processo de desenvolvimento de Software [18]

queridos [11, 4]. A partir de um esboço dos requisitos iniciais para o problema a ser resolvido, um modelo de processo de software será adotado, o qual resultará em uma aplicação para atender os requisitos dos usuários.

Um processo de desenvolvimento de software pode possuir subprocessos interrelacionados. O processo de desenvolvimento e manutenção do software pode ser visto como um subprocesso designado por Processo de Produção. Outro subprocesso, designado por Processo de Gestão, seria o controle e disponibilização de recursos necessários ao sucesso do Processo de Produção[19].

Com a tecnologia de processo de software, surgiu nos últimos anos um novo conceito de ambiente de trabalho que dá suporte de forma integrada aos processos de gestão e produção: Ambientes de Desenvolvimento de Software Orientados ao Processo ou os chamados PSEEs (*Process-Centered Software Engineering Environments*) [10], que constituem um tipo especial de ambiente de desenvolvimento de software apoiando a definição rigorosa de processos de software e objetivando a automação da gerência do desenvolvimento. Tais ambientes geralmente provêm serviços para análise, simulação, execução e reutilização das definições de processos, que cooperam no aperfeiçoamento contínuo de processos, conforme mostra a Figura 1, sendo que alguns desses PSEEs utilizam recursos de inteligência artificial para apoiar o processo de desenvolvimento de software [20].

Uma das formas de apoiar o processo do negócio é a utilização de *workflow* (WF), cujo principal objetivo é modelar e automatizar processos de negócio (*Business Process – BP*), envolvendo a administração de organizações e coordenando os processos desempenhados por profissionais diversos. Mais formalmente, segundo a *Workflow Management Coalition* (WfMC) [1], sistemas de *workflow* estabelecem "a automação total ou parcial de um processo de negócio, na qual documentos, infor-

mação e/ou tarefas são passadas de um participante para outro a partir de ações, de acordo com um conjunto de regras procedimentais".

Este artigo apresenta, na Seção 2 os conceitos fundamentais dos Ambientes de Engenharia de Software. Na Seção 3 são abordados os *workflows*. Na Seção 4 são mostradas as relações existentes entre os PSEEs e o *workflow* e como a integração destes dois conceitos podem auxiliar no processo de desenvolvimento de software. Finalmente, na Seção 5 são apresentadas algumas conclusões sobre o tema.

2 Ambientes de Engenharia de Software

Embora o uso de ferramentas para auxiliar os desenvolvedores na produção de software venha sendo aplicado já há algum tempo, o conceito de *Software Engineering Environment (SEE)* (Ambiente de Engenharia de Software) é bastante recente. Um SEE é definido como uma coleção de ferramentas que fornece apoio automático, parcial ou total, às atividades de engenharia de software. Normalmente essas atividades são executadas dentro de uma estrutura de projeto de software, e se referem a aspectos tais como especificação, desenvolvimento, reengenharia ou manutenção de sistemas de software [19].

O termo SEE pode ser aplicado a vários sistemas de alcances bem diferentes: desde um conjunto de poucas ferramentas executando sobre um mesmo sistema, até um ambiente totalmente integrado capaz de gerenciar e controlar todos os dados, processos e atividades do ciclo de vida de um software. Graças a automatização total ou parcial das atividades, um SEE pode contribuir com importantes benefícios para uma organização: redução de custos, aumento da produtividade, melhora da gestão e maior qualidade dos produtos finais. Por exemplo, automatização de atividades repetitivas, como execução de casos de teste, não apenas melhoram a produtividade, mas também ajudam a garantir o término e a consistência das atividades testadas [19]. De modo geral, um SEE gerencia informações relacionadas com:

- Desenvolvimento ou manutenção do software (especificações, dados de projeto, códigos fonte, dados de teste, planos de projeto, ...);
- Recursos do projeto (custos, recursos de informática, pessoal, responsabilidades e obrigações, ...);
- Aspectos organizacionais (política da organização, padrões e metodologias empregadas, ...).

Um SEE oferece suporte às atividades humanas mediante uma série de serviços que descrevem as capacidades do ambiente. Os serviços proporcionam uma

correspondência entre um conjunto de processos selecionados, relativos ao ciclo de vida do software, e sua automatização mediante o uso de ferramentas. Na maioria dos casos as funcionalidades de uma ferramenta estão relacionadas com um ou mais serviços.

2.1 Integração

O que mais diferencia um SEE de um simples conjunto de ferramentas sendo executadas em um computador em um mesmo sistema operacional é o grau de integração que o ambiente oferece [19]. O conceito de integração aplicado a SEE pode significar várias coisas relacionadas mas diferentes:

- O grau em que ferramentas diferentes podem comunicar eficientemente entre si dentro da estrutura de trabalho do SEE;
- Uma medida das relações entre os componentes de um SEE;
- Facilidade, interoperabilidade, portabilidade, escalonabilidade, produtividade, etc., produzida por uma integração "sem remendos" entre componentes de um SEE.

Compartilhar um mesmo sistema de gerência de objetos em vez de um sistema de arquivos separados para cada ferramenta é um aspecto importante da integração, mas não é o único. Um SEE deve disponibilizar um conjunto de interfaces que permita a cooperação arbitrária entre ferramentas de fabricantes diversos. Por esta razão, a integração implica nos três seguintes aspectos:

- **Um conjunto de serviços.** Há vários serviços que são aplicáveis à integração. Por exemplo, utilizar um sistema de gerência de objetos (SGO) comum com planos comuns, permite que as ferramentas compartilhem objetos; utilizar características de apresentação global na interface de usuários permite disponibilizar um "aspecto de visualização" semelhante em todas as ferramentas; ou os serviços de gerenciamento de processos e de comunicação são necessários de modo que as ferramentas podem comunicar umas com as outras.
- **Um novo formato para cada serviço.** Possuir serviços comuns apenas permite, não obriga a integração (os construtores de ferramentas não são obrigados a utilizá-los). Este novo formato indica o grau em que um serviço pode contribuir para aumentar a integração.

- **Uma política.** Também é necessário implantar políticas de modo que os construtores de ferramentas, frameworks de trabalho e plataformas, utilizem os serviços de integração de forma eficiente. Um exemplo disto são as "guias de estilo" para construção de ferramentas.

Na base da definição anterior, a integração é uma propriedade de relacionamento entre os componentes de um SEE. Esta relação pode ser entre ferramentas e estruturas de trabalho, ferramentas e interfaces de usuário, ferramentas e ferramentas, ferramentas e SGO, etc. Por isso a integração pode ser considerada de várias perspectivas diferentes:

- Para o usuário, um SEE integrado provê uma visão comum de sistema. O ambiente completo funciona como uma ferramenta consistente, em vez de uma coleção de diferentes operações ativadas separadamente.
- Para o desenvolvedor de ferramentas, um SEE integrado fornece uma interface consistente para construir ferramentas. Desta maneira, as funções para interação com o SGO, gerência de processos e outros serviços devem estar especificadas de forma clara, e as ferramentas poderão comunicar entre si de forma simples e segura.

Em geral, a necessidade de integração em um SEE engloba várias partes diferentes:

- **Dados.** A integração dos dados e a capacidade de compartilhar informações dentro de um SEE. O grau de integração dos dados pode ser alto (as ferramentas usam uma base de dados comum com um plano comum), médio (formatos de dados comuns) ou baixo (utilização de mecanismos de tradução);
- **Controle.** A integração do controle é a capacidade de combinar as funcionalidades oferecidas em um ambiente de forma flexível. As combinações podem se ajustar às preferências de um projeto e ser dirigidas por processos de software subjacentes;
- **Apresentação.** A integração de apresentação é a capacidade de integração com funcionalidades de ambiente através de telas de aparência semelhante e meios de interação semelhante;
- **Processos.** A integração de processos é a habilidade de adaptar as funcionalidades do ambiente utilizando um processo de software pré-definido;

- **Framework de trabalho.** Este se refere ao grau em que as ferramentas estão integradas (isto é, faz uso de) ao framework de trabalho.

2.2 PSEE

Como dito anteriormente, PSEEs [10] constituem um tipo especial de ambiente de desenvolvimento de software que surgiu nos últimos anos para apoiar a definição rigorosa de processos de software, objetivando automatizar a gerência do desenvolvimento. Tais ambientes provêem serviços para análise, simulação, execução e reutilização das definições de processos, que cooperam no aperfeiçoamento contínuo de processos.

A modelagem de processos de software não consiste apenas em escrever programas que automatizem completamente o processo de desenvolvimento de software e nem descrevem tudo o que os atores do processo devem fazer [16]. Enquanto os programas de computador são escritos para definir o comportamento de uma máquina determinística, os programas de processo são escritos para definir possíveis padrões de comportamento entre elementos não-determinísticos (atores) e ferramentas automatizadas [22]. Como consequência, um PSEE deve ainda permitir que os atores envolvidos no processo recebam orientação automatizada e assistência na realização de suas atividades, sem interferência no processo criativo [12]. Além disso, processos ainda podem ser modificados dinamicamente, em resposta a estímulos organizacionais ou mudança nos requisitos do software em desenvolvimento.

A literatura especializada define três tipos principais de modelos [6, 3, 5]:

- **Modelos Abstratos** (patterns ou templates), que fornecem moldes de solução para um problema comum, em um nível de detalhe que idealmente não está associado a uma organização específica. Um processo abstrato é um modelo de alto nível que é projetado para regular a funcionalidade e interações entre os papéis de desenvolvedores, gerentes, usuários e ferramentas em um PSEE [23];
- **Modelos Instanciados** (ou executáveis) são modelos prontos para execução, podendo ser submetidos à execução por uma máquina de processo. O modelo instanciado é considerado uma instância de um modelo abstrato, com objetivos e restrições específicos, envolvendo agentes, prazos, orçamentos, recursos e um processo de desenvolvimento;
- **Modelos em Execução** ou **Executados** registram o passado histórico da execução de um processo,

incluindo os eventos e modificações realizadas no modelo associado.

A arquitetura de um PSEE define como componente central a máquina de processo [3] que auxilia na coordenação das atividades realizadas por pessoas e por ferramentas automatizadas, sendo responsável pela interpretação/execução dos modelos de processos descritos com PMLs (*Process Modeling Languages*). Uma máquina de processos é responsável por: ativar automaticamente atividades sem intervenção humana por intermédio de uma integração com as ferramentas do ambiente; apoiar o envolvimento cooperativo dos desenvolvedores; monitorar o andamento do processo e registrar o histórico da sua execução [10]. A máquina de processo também deve garantir a execução das atividades na seqüência definida no modelo de processo; a repetição de atividades; a informação de *feedback* sobre o andamento do processo; a gerência das informações de processo (incluindo gerência de versões); a coleta automática de métricas; a mudança do processo durante sua execução; a interação com as ferramentas do ambiente e a gerência de alocação de recursos [10].

O mecanismo de execução de processos de um SEE pode conter diversas instâncias simultâneas de máquinas de execução. Isto é necessário porque o SEE pode estar sendo utilizado para desenvolvimento em diversos projetos em uma organização. Portanto, de forma geral um mecanismo de execução consiste de uma ou várias máquinas de execução. As máquinas de execução possuem componentes que trabalham na execução de processos e na integração com o restante do ambiente, pois a execução envolve desde a interface com o usuário até a gerência dos objetos no banco de dados. Dentro de um PSEE, o mecanismo de execução pode ser tratado como mais uma ferramenta ou pode ser um componente básico do ambiente.

3 Workflow

As tecnologias de *workflow* (WF) têm sido utilizadas para modelar, melhorar e automatizar processos de negócios (PN). Um processo de negócio possui um conjunto de procedimentos ou atividades interligadas e sua execução visa a alcançar um certo objetivo ou meta relacionada ao contexto de uma organização. Um *workflow* pode ser visto como a automação total ou parcial de um processo de negócio e esta automação consiste em representar um *workflow* em um formato compreensível por uma máquina. Embora a modelagem e execução de *workflow* sejam relativamente novas, muitos dos seus conceitos já existem há algum tempo e têm sido aplicados a diversas áreas [19].

Os chamados Sistemas de Gerência de *Workflow* (*Workflow Management System* - WFMS) provêm um conjunto de ferramentas de software de apoio para a definição e execução de *workflows*. Os WFMS surgiram primeiro na indústria e só depois se tornaram alvo de pesquisa. Em função disto, podemos observar um grande número de produtos nessa área que resolvem problemas concretos, mas que, em muitas situações, não apresentam uma base científica forte que permita resolver certas limitações. Além disso, existem diversos tipos de WFMS para os diversos tipos de negócio de *workflow*. Nem todos eles oferecem os mesmos benefícios na hora de modelar ou executar um *workflow*.

Com tudo isso o campo dos WFMS passou por um momento onde pode ser observada uma carência de padronizações. A partir daí, surgiu uma organização internacional denominada *Workflow Management Coalition* (WfMC) cujo objetivo é unir esforços para propor padrões no campo de WFMS. A criação dessa organização foi interpretada pelos desenvolvedores como um avanço nessa área. Foram propostos padrões que especificam um modelo básico para a definição de um *workflow* até que outras definições mais complexas fossem propostas.

De acordo com o WfMC, um WFMS é um sistema que define e organiza a execução de *workflow* pelo uso de software [8], sendo capaz de interpretar a definição de um processo, interagir com os participantes e invocar o uso de ferramentas e aplicações quando necessário. Podemos observar que um WFMS desempenha duas tarefas distintas relacionadas entre si. A primeira tarefa é a especificação do *workflow*. Nesta tarefa é produzida a definição do *workflow* representando o processo de negócio. A outra tarefa é a animação ou execução do *workflow* [19].

Na Figura 2 pode-se observar as duas tarefas desempenhadas por um WFMS. O WFMS é um conjunto de atividades executadas numa dada ordem. Essas atividades podem ser executadas seqüencialmente, em paralelo, ou uma entre várias outras atividades possíveis. As atividades contêm parte do trabalho de um processo e podem requerer a participação de agentes humanos ou invocar aplicações, levando totalmente ou parcialmente o trabalho da atividade para ser executado externamente.

Para definir um *workflow*, é necessário especificar as atividades e seus componentes na ordem em que devem ser executados. Um *workflow* é representado por um grafo dirigido onde os nodos são as suas atividades e as setas indicam a precedência de execução. A maioria desses WFMS incluem, entre seus componentes, um editor gráfico que permite representar as atividades,

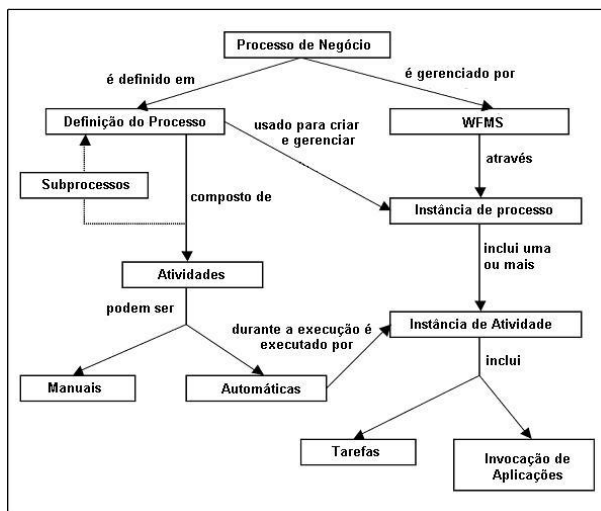


Figura 2: Tarefas desempenhadas por um WFMS [19]

o controle de fluxo entre elas, os fluxos de dados e os participantes em cada atividade.

A execução de um *workflow* em um WFMS consiste na criação, manipulação e destruição de cópias ou instâncias dos processos. Cada instância do processo terá uma identidade visível externamente e um estado interno que representa o progresso em relação ao estado final e seu estado com respeito às atividades que o compõem. Sempre que, na execução do processo, houver a necessidade de invocar uma atividade, será criada uma instância da mesma. É ela que se encarrega de executar as ações definidas, processar os dados que são necessários e invocar a aplicação externa correspondente no caso de se tratar de uma atividade automática. Toda a execução de um *workflow* é realizada de acordo com uma especificação prévia.

A WfMC caracteriza um WFMS como um sistema que dá apoio em três áreas funcionais:

- **Área de funções em tempo de construção:** onde é considerada a modelagem de processos;
- **Área de funções de controle em tempo de execução:** onde é considerada a gerência de processos que são executados;
- **Área de interações em tempo de execução:** onde são considerados os usuários humanos e as ferramentas de aplicações de tecnologia da informação. A WfMC propôs um modelo de referência de WFMS baseado num modelo genérico que identifica seus componentes e suas diferentes interfaces, que permitem a comunicação entre os diferentes níveis [8].

3.1 Ciclo de vida de um *Workflow*

O desenvolvimento e uso de um *workflow* engloba uma série de atividades que devem ser realizadas numa seqüência determinada. O ciclo de vida de um *workflow* (WFLC - *Workflow Life Cycle*) define como essas diferentes atividades são ordenadas em cada momento, assim como os fluxos de informações que constituem as entradas e saídas de cada atividade [2]. O WFLC inclui três processos centrais:

1. **Construção e estabilização** (passos 2 e 3 da Figura 3): Partindo da modelagem de negócio produzida na primeira fase¹ de análise do processo de negócio, uma linguagem gráfica é usada para modelar o *workflow* usando conceitos tais como processos, atividades, tarefas, etc. Como ocorre com qualquer produto de software, o processo de construção de *workflow* não é livre de erros, por isso é desejável que seja feita a detecção de erros antes de passar para o próximo passo. Uma prototipagem de processo iterativa permite simular execuções de *workflow* para detectar e corrigir erros de especificação.
2. **Depuração** (passos 4, 5, 6 e 7 da Figura 3): Uma vez que uma versão estável do processo de negócio está disponível como uma especificação de *workflow* com todas as suas características, uma versão operacional é gerada pelo mapeamento de conceitos modelados em uma representação executável. Como resultado da execução do *workflow*, é criado um registro onde são colocadas as informações sobre a execução do processo: tarefas executadas, início e término das tarefas executadas, recursos usados, etc., que podem ser analisadas posteriormente para detecção de anomalias (*deadlocks*, gargalos, etc) causadas por um projeto de *workflow* mal feito. Uma vez que uma anomalia é detectada, deve ser feita uma revisão do projeto. Note que o objetivo da depuração durante todo o processo é a melhora da qualidade do projeto do *workflow*.
3. **Remodelagem do processo de negócio** (passos 8, 9 e 1 da Figura 3): A partir dos dados obtidos pelo uso do registro de execução, é possível verificar a adequação do modelo do processo de negócio às

¹A separação entre análise do processo de negócio e modelagem do *workflow* é deliberada da seguinte forma: já existe uma modelagem do negócio em uma determinada empresa quando ela resolve usar WFMS; entretanto, se este não for o caso, a modelagem do processo de negócio e modelagem do *workflow* devem ser construídas juntas.

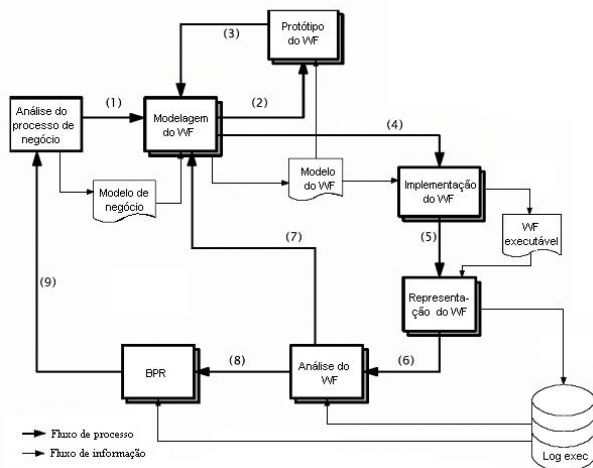


Figura 3: Ciclo de Vida de um *workflow* [2]

metas da organização. Caso não esteja correto, é obrigatória a revisão do modelo. A fase de remodelagem do processo de negócio (BPR - *Business Process Reengineering*) conduz novamente à fase de análise do processo de negócio, onde o modelo é revisado. A atualização do modelo inicia uma nova iteração do processo de estabilização e depuração.

4 Aplicação de *workflow* como suporte a processos de software em PSEE

Como foi visto na seção 2.2, a gestão de processo é um dos serviços básicos de um PSEE. Ela ajuda os projetos de software a alcançar disciplina, controle e entendimento claro dos seus processos e atividades [17]. Um processo de software pode ser visto como um caso particular de um processo empresarial a medida que engloba um conjunto de atividades ordenadas que têm como objetivo o desenvolvimento ou manutenção de software. Desta forma, a tecnologia de *workflow* pode ser utilizada para dar apoio ao processo de software em um PSEE.

A Gerência de Processo envolve os seguintes serviços cujas descrições são especificadas pelo padrão ISO 15940:

1. **Definição de processos:** oferece apoio à implantação dos processos organizacionais, cobrindo o ciclo de vida do software pela adaptação e particularização de um conjunto de classes de processos de referência de alto nível. Essa adaptação inclui as políticas específicas da organização e sua infraestrutura, métodos e procedimentos.

2. **Biblioteca de processos:** oferece apoio para reutilizar as capacidades de processos com base nos seus ativos ou recursos. Um recurso de processo pode ser tão simples quanto a definição de uma tarefa simples ou tão complexo como a definição de um ciclo de vida completo.
3. **Inicialização de processos:** oferece apoio ao estabelecimento de um modelo de ciclo de vida, ao estabelecimento de um grupo de processos e ao estabelecimento de especificações para satisfazer os requisitos e as restrições de um projeto particular.
4. **Uso de processos em projetos:** inclui as capacidades para ajudar a utilizar processos dentro de um projeto.
5. **Supervisão de processos:** oferece apoio à observação, descoberta, aparecimento e inclusão de atividades de processos dentro de projetos.
6. **Melhoria de processos:** oferece apoio à avaliação, medição e modificação dos processos organizacionais e de projetos, e dos ciclos de vida dos projetos.
7. **Documentação de processos:** oferece apoio a todos os outros serviços para a documentação do processo.

A Gerência de Processos pode ser auxiliada por sistemas de *workflow* cujo elemento básico é o processo, além de sua composição e as pessoas que o executam. Não há restrições às metodologias ou ferramentas específicas para a modelagem e execução de um *workflow*. Essa flexibilidade favorece a definição de processo de software mais específico a cada contexto de desenvolvimento e contribui para a evolução desses processos em relação às modificações.

Como já vimos na Seção 1, os WFMS disponibilizam um conjunto de ferramentas de software que possibilitam definir e executar um *workflow*. Um WFMS fornece apoio a processos de software, e dentre as suas potencialidades estão [19]:

- **Modelagem de processos:** WFMSs oferecem linguagens de programação de alto nível para definição de processos. Em sua maioria, possuem editores gráficos que facilitam a definição dos modelos. A modelagem de processo tem uma importância fundamental, pois, representa a base para a automação do processo.
- **Separação entre modelo de processo e suas instâncias em execução:** WFMSs separam a definição de processo da sua execução permitindo uma

maior flexibilidade para a realização de modificações. Desta forma, o processo de software pode ter as suas definições modificadas com facilidade, sem afetar as instâncias de processos em execução ou até refletir essas alterações diretamente nessas instâncias que passam a seguir a nova definição.

- **Heterogeneidade:** WFMSs são sistemas projetados para operar em ambientes de informação modernos. Esses sistemas estão preparados para a integração com outros ambientes de *workflow*, permitindo que os processos possam ser executados por serviços de *workflow* diferentes e integrados através de uma interface previamente definida. Em função de sua característica de Heterogeneidade, os WFMSs oferecem a possibilidade de operar dentro de ambientes de informação distribuídos e heterogêneos, o que é útil quando falamos em projetos de construção de software multidisciplinares, envolvendo diversos setores de uma organização ou mesmo organizações diferentes com agentes distribuídos em lugares distantes.
- **Execução de processo:** A execução de processos em WFMSs corresponde à instanciação dos processos e suas atividades. Em contextos de desenvolvimento de software, essa forma de execução auxilia os participantes do processo a entenderem seu papel dentro do processo e a visualizar como está o andamento deste processo. Além disso, auxilia os participantes a coordenarem suas atividades individuais, sendo notificados sobre as atividades que esperam sua atuação. Tornam-se mais explícitas as responsabilidades de cada participante com as atividades de desenvolvimento e com as atividades individuais, contribuindo assim para o bom andamento do processo.
- **Supervisão:** WFMSs oferecem facilidade de coordenação, supervisão e auditoria do processo, provendo diversas informações sobre o seu andamento. Este pode ser um recurso inicial para a identificação de pontos a serem melhorados dentro do processo.
- **Apoio e cooperação:** WFMSs demonstram uma grande preocupação com os aspectos de colaboração entre os grupos de trabalho no processo. Esses ambientes concentram-se no apoio a todo o processo no que diz respeito a suas atividades e ao controle das atividades em relação ao acesso aos dados armazenados.
- **Reconhecimento do processo de trabalho:** WFMSs incorporam o usuário no processo fazendo uma in-

teração menos passiva. Isto é bastante útil para apoiar a personalização dos trabalhos individuais, oferecendo maior liberdade para a gerência de responsabilidades, além de auxiliar os executores do processo a reconhecer suas atividades dentro do mesmo.

5 Conclusão

Como se pode observar nesse trabalho, existe uma série de deficiências ligadas ao desenvolvimento de software, uma vez que os softwares a serem desenvolvidos são cada vez mais complexos. Baseado nesse contexto, foram abordados alguns conceitos que visam à melhoria da produtividade no desenvolvimento de software para a obtenção do aumento da qualidade do software produzido.

O objetivo deste trabalho foi discutir conceitos sobre ambientes de desenvolvimento de software orientados ao processo, gerência de processos, *workflow* e a integração entre esses conceitos visando obter uma melhoria no desenvolvimento e manutenção de softwares.

A tecnologia de processo de software, quando auxiliada por gerência de processos e *workflow* pode trazer inúmeros benefícios para o aumento da produtividade e da qualidade do software produzido. Esses benefícios podem ser visualizados não só pela parte organizacional, como também pelos desenvolvedores e pessoas incluídas no processo de desenvolvimento de software.

Referências

- [1] Workflow management coalition. <http://www.aiim.org/wfmc>, 1999.
- [2] CANÓS, J. H., C., PENADÉS, M^a C. From software process to workflow process: the workflow lifecycle. September 1999.
- [3] DERNIAME, J., W., KABA, B. Software process: Principles, methodology and technology. *Springer*, 1999.
- [4] DOWSON, M., R., NEJMEH, B. Fundamental software process concepts. *AICA Press*, May 1991.
- [5] FEILER, P. H., H. Software process development and enactment: Concepts and definitions. *IEEE Computer Society Press*, May 1993.
- [6] FRANCH, X., R. Supporting process reuse in promenade. February 2002.

- [7] GIMENES, I. O processo de engenharia de software: Ambientes e formalismos. *XIV Congresso da Sociedade Brasileira de Computação*, 1994.
- [8] HOLLINGSWORTH, D. Workflow management coalition - the workflow reference model. *Document Number TC00 - 1003 - WfMC*, 1995.
- [9] HUMPHREY, W. S. The software engineering process: Definitions and scope. *IEEE Computer Society Press*, 1989.
- [10] LIMA, C. Um gerenciador de processos de software para ambiente prosoft. Master's thesis, CPGCC-UFRGS, 1998. Dissertação de Mestrado.
- [11] LONCHAMP, J. A structured conceptual and terminological framework for the software process engineering. *IEEE Computer Society Press*, May 1993.
- [12] NGUYEN, M., W. Total software process model in epos. <http://www.idt.unit.no/epos/Papers>, 1997.
- [13] OSTERWEIL, L. Software process are software too. *International Software Process Workshop*, 1987.
- [14] PAULK, M. *The Capability Maturity Model: Guidelines for Improving the Software Process*. Addison-Wesley, 1994.
- [15] PRESSMAN, R. S. *Software Engineering: A Practitioner's Approach*, volume 5. McGraw-Hill, 2001.
- [16] REIS, R. Q., N. Uma avaliação dos paradigmas de linguagens de processo de software. Março 1999.
- [17] REIS, R. Q., N., REIS, C. A. L. Automação no gerenciamento do processo de engenharia de software. *EIN - Escola de Informática Norte*, 2002.
- [18] RUIZ, F. *MANTIS: Definición de un Entorno para la Gestión del Mantenimiento de Software*. PhD thesis, Dep. de Informática, Universidad de Castilla-La Mancha, 1998. Tesis Doctoral.
- [19] SATLER, M. F. Utilidad de la tecnología de workflow en los psees. Dep. de Informática, Universidad de Castilla-La Mancha, 2004.
- [20] SILVA, R. A. C. Inteligência artificial aplicada a ambientes de engenharia de software: Uma visão geral. *INFOCOMP Journal of Computer Science*, 4(4):27–37, 2005.
- [21] SILVA, R. A. C., BRAGA, J. L., SILVA, C. H. O., and SOARES, L. S. Pspm na melhoria da competência pessoal no desenvolvimento de software. In *Proceedings of JIISIC'06*, Puebla, Puebla, Mexico, 2006.
- [22] TULLY, C. Representing and enacting the software process. June 1989.
- [23] WANG, Y., K. Software engineering processes: Principles and applications. 2000.