

Frameknow – Um framework para gestão de processos de Software assistida por simulação baseado em padrões

Dawilmar Guimarães Araújo^{1,2}, Sólon Venâncio Carvalho¹, Nilson Sant'Anna¹

¹Laboratório de Computação e Matemática Aplicada
Instituto Nacional de Pesquisas Espaciais (LAC/INPE)
Caixa Postal 515 – 12.245-970 – São José dos Campos – SP – Brazil

²Universidade de Taubaté (UNITAU)
R. Daniel Danelli S/N – 12.031-000 – Taubaté – SP – Brazil

dawilmar@lac.inpe.br, nilson@lac.inpe.br, solon@lac.inpe.br

Abstract. *The Process Machines are technologies treated as the core of Process-centered Software Engineering Environment. These environments are dynamic, allowing the occurrence of incomplete data and uncertainty to become present in decisions. The objective here is to describe the Frameknow: a complementary look to Process Machines based on process extracted patterns and in interplay real and simulated data context for treating such problems. Cases of studies and lessons learning are discussed in the context of a experience of prototype.*

Resumo. *Ambientes de Engenharia de Software são dinâmicos, o que torna não incomum a ocorrência de problemas como dados incompletos, estimativas não realistas, baixa eficácia e incerteza nas decisões e nas práticas de seus gestores. Diferentes alternativas de soluções são investigadas a fim de se reduzir estes resultados e dar suporte a estes ambientes. Este trabalho oferece o Frameknow; uma abordagem complementar de uma solução baseada na combinação de diferentes conceitos envolvendo simulação, extração de padrões das bases dos processos, um jogo de dados entre o contexto real e virtual. Casos de estudos e lições aprendidas são discutidos no contexto de uma experiência do protótipo.*

1. Introdução

“A qualidade do processo influencia a qualidade do produto” (Kellner, 1999).

Ambientes de Engenharia de Software (AES), caracterizam-se por uma particular dinâmica onde se observam: não ser incomum a falta de dados ou ocorrência de dados incompletos (ou imprecisos), a incerteza nas decisões de gestores, realização de estimativas mais acuradas, dificuldade em estabelecer e reutilizar processos e prover uma efetiva e continua melhoria destes etc (Silva, 2006), (Reis, 2003).

Por outro lado, problemas como a falta de ferramentas adequadas e nível de experiência de gerentes são continuamente apontados como as principais causas de insucesso dos projetos de software (Valente e Falbo, 2002) e (Sant'Anna, 2001).

Tal fato, discutido por Armbrust (2003), sustenta-se na tese de que a

instrumentação da organização deva ser usada a fim de reduzir (ou eliminar) estes problemas. Acrescenta ainda aquele autor, que os esforços provenientes de se adquirem e manterem conhecimentos e prover recursos para analisar e melhorar os processos pertinentes são incipientes.

Diferentes soluções são direcionadas e adotadas a fim de se reduzir este cenário, podendo destacar: a definição e adoção de normas e conjuntos de referências de práticas para a maturidade e disciplina das organizações como, por exemplo, CMMI¹ (*Capability Maturity Model Integration*), MPS.Br² (Modelo de melhoria dos Processos de Software Brasileiro), PMBoK³ (*Project Management Body of knowledge*); adequação da infra-estrutura, regras e modelos organizacionais para abordagens direcionadas ao gerenciamento do ambiente orientado para processos. Segundo Reis (2003), estes esforços definem os Ambientes de Engenharia de Software Centrado em Processos (ou PSEE-*Process-centred Software Engineering Environment*).

Muitas das alternativas de soluções citadas anteriormente são apoiadas por Tecnologias de Processos de Software (TPS) e entendidas como núcleo dos PSEE. Estas se apresentam com os mais variados e diferentes recursos (ou funcionalidades), e vem sendo implementados suportando as práticas convencionais da gestão de processos, as práticas oferecidas pelos guias e normas supracitados e oferecendo diferentes visões dos processos adotados considerando o mesmo ambiente de ferramenta, conforme é corroborado por Silva (2007), Armsbrust (2003), Reis (2003) e Silva (2001).

Entretanto, observa-se nas TPS a falta de funcionalidades que provêem subsídios para lidar com uma maior qualidade dos dados, ou com a ausência ou imperfeição destes, apoio os gestores para lidar com experiências adquiridas, suporte a ações da gerência como aferição de estimativas e acompanhamento da execução dos processos por antecipação de resultados etc.

1.1. Objetivos

Este trabalho apresenta o *Frameknow*; uma estrutura que implementa o conceito de gestão assistida por simulação baseado em padrões de processos, cujo objetivo é apoiar a gestão de projetos e processos de *software*, colaborar no sentido de reduzir o cenário de problemas exposto e ao mesmo tempo ampliar as possibilidades de uso recursos das TPS conforme discutidos na seção anterior.

2. Base conceitual

Esta seção resume aspectos das bases teóricas que envolvem o *Frameknow*, a saber: Processo de Software, Tecnologias dos Processos de Software (TPS), Simulação de processos e gerenciamento do Conhecimento Organizacional (CO).

2.1. Processo de software

Segundo a ISO15504, um processo de software é definido como uma coleção de atividades inter-relacionadas capazes de realizar a construção de um software.

¹ <http://www.sei.cmu.edu/cmmi>.

² www.softex.br

³ www.pmi.org

Uma boa definição de processo de software deve considerar aspectos de negócio da organização que o adota. Para tanto, a importância do processo de software é percebida quando integrado aos processos de negócios da organização e refletida na composição de elementos e de explícitos relacionamentos. Desta forma, uma atividade é composta por um conjunto de elementos e relacionamentos, que se define por suas tarefas, que comumente são regidos por regras e metas da organização e do próprio processo (restrições). Competências ou papéis são escritos e definidos para a realização das tarefas através de seus atores. Atores produzem produtos (objetivos) do processo ao exercerem seus papéis, executando suas tarefas, e para isto necessitam consumir recursos .

2.2. Tecnologias de Processo de Software (TPS)

Segundo Reis (2003), as TPS surgiram como resultado das oportunidades e necessidade de se produzir software como ferramental de apoio as práticas de gestão de processos. Seus principais objetivos podem ser sintetizados em: i) Definir ferramentas para descrever os processos e acompanhar a sua execução; ii) Facilitar a adoção de uma estratégia de melhoria da maturidade; iii) Permitir o registro do conhecimento produzido acerca dos processos bem sucedidos; iv) Proporcionar um controle preciso na alocação e no consumo de recursos; e v) Coletar métricas e torná-las disponíveis para consultas posteriores.

Para cada item citado no parágrafo anterior, diversas propostas podem ser encontradas na literatura (Silva, 2007) (Souza, 2001), (Reis, 2001) (Travassos, 1999), o que induz no que parece ser procedente dar créditos ao esforço da padronização, a fim de promover disciplina, produtividade e maturidade as propostas de ferramentas e as funcionalidades das ferramentas de gestão de processos de software.

2.3. Simulação de Processo de Software

A simulação de processo de software é abordada de forma geral em três clássicos modelos: evento-discreto, contínuo e híbrido [Kellner, 1999]. Independentemente da escolha do modelo de simulação, os desenvolvedores devem possuir as duas principais habilidades [Ioana, 1999]: i) de escrever modelos e reescrever em ambientes de simulação (conhecer o processo de simulação, a ferramenta utilizada e tudo que os cerca); e ii) de conhecer processos de desenvolvimento de software (aspectos fundamentais, organizacionais, negócios e tudo mais que os cerca).

É vasta a diversidade de propostas que contemplam o uso de simuladores para soluções de problemas e apoio aos processos de software. Kellner (1999) destaca o papel da simulação para a modelagem, planejamento e avaliação de processos, treinamento de equipes, previsão de qualidade de produto etc.

2.4. Conhecimento Organizacional (CO)

O Conhecimento Organizacional (CO) na maioria das vezes, observando o lado da organização, é descrito e explícito, armazenado por meio de documentos, relatórios. Por outro lado, por parte dos usuários, é do tipo tácito, relacionado e dependente das habilidades de seus proprietários, formado e caracterizado fortemente por serem indeterminados e de difícil manutenção. [Olival Jr e Menezes, 2000].

Observando Valente *and* Falbo (2002), Reis (2003), Scacchi (1999), Silva

(2001) e Souza (2001), o CO é diferentemente modelado e tratado. Encontra-se por meios formais, matemáticos, estruturados, em notação de linguagem específica, servindo mais restritivamente ao contexto limitado do problema, e para o desenvolvedor que os formulou do que a organização e usuários em geral. Servindo assim, mais restritivamente para construir novos conhecimentos, colaborar com as práticas organizacionais, ou apoio a tomada de decisão e suporte a capacidade de produzir cada vez e melhor, a qualidade dos processos etc.

Pouco tem sido mencionado sobre os instrumentos tecnológicos da gestão do conhecimento nos PSEE, ressaltando algumas citações como Falbo (1999), Reis (2001), Lima (2000), Silva (2001) e Souza (2007). Estes últimos tendo como base o trabalho de Mi e Scacchi (1999), onde é citado como pioneiro na proposta de ambiente de processos de software. E em outro caso, Araújo *and* Borges (2002) discutem as máquinas de *workflow* ou Máquinas de Processos (também vistas como TPS) como instrumentos para manutenção do conhecimento, uma vez que facilmente detectada similaridade entres estas áreas.

Diante disto, considerações acerca do uso combinado de tecnologias de gestão de conhecimentos e de processos parecem ser uma alternativa e oportunidade de proporção bastante razoável. Entendendo uma vez as relações exposta no Quadro 1.

Quadro 1. Estrutura do relacionamento da gestão do CO x Gestão de Processos no PSEE. Fonte: Adaptado de Olival Jr. e Menezes (2000).

Gestão do conhecimento	Produtos (ligação)	Gestão de Processos
Planejamento organizacional	Conhecimento (dados, informações)	Definição do processo (Processo definido e instanciado)
Instanciação e execução de métodos, técnicas e ferramentas	Maturidade (práticas definidas)	Execução do processo (Processo executável)
Avaliação de resultados organizacionais	Resultados (métricas e medidas)	Medição do processo (atributos)

3. *Frameknow*

Frameworks são propostos e diferentemente definidos como soluções para domínios de problemas específicos. Silva (2007) reflete acerca da solução tecnológica proveniente da definição e contexto de construção e da implementação de *frameworks*.

O *Frameknow* é apresentado e implementado a partir de um conjunto de conceitos como um *framework* tipo *graybox* para as funcionalidades que se propõe agregar às TPS, conforme citado na seção 1. Esta combinação de conceitos, permite sintetizá-lo em um só: “uma concepção clara do mundo real deve ser replicada em um mundo virtual de tal forma que permite uma continuidade daquele primeiro na forma do segundo, acompanhando a realidade, transformando-a e sendo transformado” [Zelkowitz *and* Wallace, 1996].

Desta forma, de novos e “velhos” conceitos, *Frameknow* apóia aspectos das práticas gerenciais, medição e análise de aspectos dos processos, simulação, extração de padrões de conhecimento da base de processos da organização, interação entre usuários e a máquina e integração mundo real e virtual.

O *Frameknow*, proposta deste trabalho, adota padrões do mercado (linguagem,

plataforma, banco de dados, protocolos de serviços etc). A adoção da estrutura é discutida mais amplamente, bem como benefícios e seus procedimentos em outro trabalho, reservando alguns aspectos gerais aqui neste artigo (Araújo, 2008).

3.2. Arquitetura

Afim de investigar a viabilidade da proposta foi adotado o ambiente de gestão de projetos livre (*dotProject*). Segundo Salvador (2005), Magalhães (2007), Hauck, Wangenheim e Thiry (2007), Nogueira e Moreira (2007) e Travassos (2007) este ambiente demonstra aderência da ferramenta a diferentes especificações de serviços para processos, principalmente como algumas descritas nos modelos de gestão de processo para maturidade das organizações MPS.Br e CMMI e para o gerenciamento de projetos baseado em processos.

A implementação do protótipo parte da construção e adequação do *framework* proposto no ambiente de gestão *dotProject*, constando dos elementos apresentados na Figura 1. Nesta estrutura cada módulo pode ser instanciado e utilizado isoladamente, no entanto principalmente para agir (ser invocado), quanto a interação entre os módulos e integração ao PSEE, ela se dá por meio de transição de dados e controle de serviço realizado pelo usuário e pelo próprio núcleo do *Frameknow*.

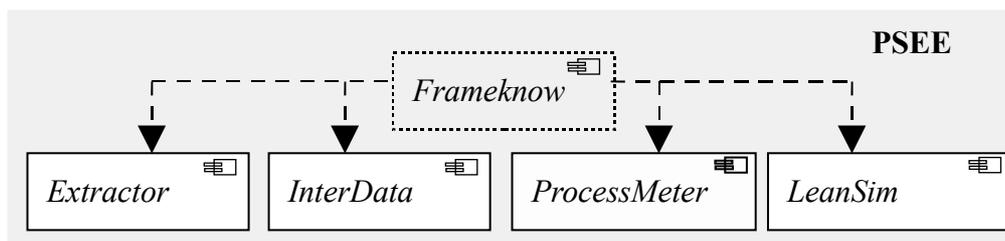


Figura 1. Componentes do *Frameknow*

3.2.1. *Extractor* – Identificador e interpretador de padrões

Extraí partes dos modelos de processos que gera uma representação (parte do processo original) para o contexto do *Frameknow*. Isto é feito utilizando-se *Add-on* que manipula padrões em um *schema* específico em linguagem baseada na notação XPDL⁴ ou por dados relacionais, por exemplo, utilizando algoritmos de *Associação*.

O *Frameknow* possui classes específicas que propicia o recurso de geração automática e autônoma do repositório de processos para subsidiar a análise dos elementos de interesse e aspectos indicados pelos usuários. Esta base é entrada para demais componentes de forma padronizada no contexto do *Frameknow*. Para futuras implementações do *Extractor* cita-se a capacidade de identificar e lidar com parâmetros imprecisos, usando conceitos de IA como crenças, opiniões etc.

3.2.2. *LeanSim* – Simulação enxuta

O *LeanSim* é um simulador flexível. O *LeanSim* implementa um conceito da simulação enxuta, para simular o modelo ou mesmo o segmento (trecho) do processo já definido no repositório de processos, aspectos dos elementos dos processos que já é instanciado e

⁴ XPDL, acrônimo para XML-Process Definition Language. Disponível em <http://www.wfmc.org>

utilizado pela organização. O *LeanSim* utiliza paradigmas evento-discreto, e para futuras implementações conceitos que combinam outros paradigmas tendem a ser usados, uma vez que o trabalho original (Araújo, 2008) propõe a implementação de um novo conceito para a gestão assistida por simulação baseado no acompanhamento convectivo de processos.

3.2.3. *ProcessMeter* – Medidor de processos

O módulo *ProcessMeter* trabalha com métricas diretas e indiretas baseado no conceito de construção ontológica de métricas. Quanto a métrica direta, é estabelecido funções internas. Já na métrica indireta, tem-se a definição por conta do usuário, provendo suas particularidades. É esperado para o *Frameknow* futuramente ser estendido e atender a um conjunto maior de métricas, associado ao aspecto de interesse do gestor.

3.2.4. *InterData* - Jogo de dados da interatividade real e máquina

O módulo *InterData* utiliza o conceito de combinação entre dados recuperados na base de processos e de dados provenientes da matriz de opiniões de especialistas para fornecer condição de tratamento de dados no contexto do PSEE. Esta funcionalidade busca suprir deficiência de falta de dados, apoiar estimativa e aferição destas, gerar modificações nas simulações propiciando cenários para análise, avaliação e melhorias dos processos.

A diferença entre a opinião do usuário e do dado recuperado, tratado e devolvido pelo *Frameknow* contribuem para a aferição que deve ser realizada nas próximas estimativas do próprio especialista e de correção de parâmetros para o simulador. Este módulo reproduz uma interpretação da capacidade esperada do processo e da participação dos elementos do processo envolvido (ator, recurso, tarefa etc), permitindo visualizar oportunidade de melhorias.

4. Casos de estudos

Considerou-se para fins de “lições aprendidas” dois diferentes casos:

a) Caso 1 – trata de base de processos empíricos de uma prestadora de serviços de assistência tecnológica, sem nenhum tipo de organização (como nível 0 do CMMI), com alguns poucos registros realizados. O projeto em estudo é implementação de um módulo de software em sistemas de catracas eletrônicas de uma empresa de ônibus da cidade de Jaguariúna. Este caso conta com 2 funcionários que assumem quase todos os papéis e com algumas contratações sazonais.

b) Caso 2 – do grupo de pesquisa do convênio Sagem-Orga UNITAU para desenvolvimento de software especializado em cartões (*SmartCart*). O grupo por parte da escola tem em média por 8 alunos de graduação, 2 professores e 1 funcionário da empresa conveniada. Utilizam diferentes metodologias e tecnologias para apoiar práticas de engenharia de software fundamentais (*Extreme Programming-XP*) e a gestão do projeto (*ad-hoc*) com o *MS-Project*[®] e *GanttProject*[™]. Foram considerados dois projetos (*PSC-Personal Services Card* e *CKM-Cryptografic Key Management*).

Optou-se para os casos de estudos, observa ocorrências de registros de três ativos dos processos (tarefas, atores e artefatos). Também considerados alguns

indicadores métricos de capacidade do processo sob algumas métricas diretas realizadas, estimadas e simuladas a respeito destes ativos como: tempo e atraso, para tarefas; produção e retrabalho (teste e correção) para artefatos; experiência, treinamento e dedicação e a métrica indireta de produtividade para atores.

4.1 Resultados preliminares e Lições aprendidas

Da aplicação do protótipo, são apresentados pontos de melhoria de gerenciamento, capacidade para desempenho e análise de processos, com base nas metas organizacionais traçadas.

Foram considerados dados incompletos e após alguns exercícios pode-se contabilizar os casos recuperados (Tabela 1, em valores arredondados), O percentual recuperado e tratado para cada caso 13%, 75% respectivamente para os casos 1, 2 para os projetos (1 e 2). Com as bases de dados adotadas para estudos e devidamente tratadas

	Volume de Dados			Volume de Dados (% da base)		
	Base	Incompletos	Recuperados	Incompletos	Recuperados	Tratado
Caso 1	150	113	15	75%	10%	13%
Caso 2	570	120	90	21%	16%	75%

Tabela 1 – Resultados das experiências realizadas para cada caso

A menor das parcelas da Tabela 1 se deu por falta completa de registros e opiniões especialistas, uma vez que o papel do “Gerente de Projeto” não existe. O caso 2, e maior fator de recuperação, se explica pelo fato de que aquela organização possui uma sistemática de registro já instalada e procurar dar respostas as questões tipicamente úteis a eles e aos programas que futuramente lhes servirão de base para novos contratos. O papel de “Gerente de projeto” existe e participa fielmente com grande interesse em praticas de engenharia de software. Entre os projetos do caso 2, o menor em sucesso de recuperação, explica-se pela falta de acesso a conjunto de dados que por ventura possam contribuir para estas análises, uma vez este já tenha sido totalmente realizado.

Em função do número e relação de dependência das atividades (seqüenciais), das equipes pequenas e com atribuições mais bem definidas, caso 2, pode-se recuperar informações de capacidade de pares para trabalhos específicos e produtividade, associadas ao tempo de experiência/treinamento. A equipe trabalha colaborativamente nas fases de projeto (*design*) e com o conceito de “pares de 2” da metodologia XP para desenvolvimento e testes, alternando-se sempre entre si de forma livre e intuitiva.

Durante a aplicação, não foram significativos os estudos para o caso 1, uma vez que não puderam ser replicadas e apenas confirmadas em potenciais melhorias de prática de gestão. As estimativas realizadas nos casos rederam considerações acerca do entendimento da complexidade envolvida, dependência de outros fatores. No caso 2, as questões de afinidade de grupo e estabilidade de horários de trabalho corresponderam a uma diferença menor entre o estimado, simulado e realizado. Tanto para o caso 1 quanto o caso 2, todos os membros defendem que as paradas formais entre horários de trabalhos devem ser freqüente, estáveis e conhecidas. O que poderá refletir em diferentes particularidades de cada grupo quanto a adequação de modelos de trabalhos.

Segundo os responsáveis das equipes e usuários de cada caso, um significado prático para este estudo (mesmo em estagio inicial), foram os ganhos provenientes do aumento da visibilidade da execução dos processos oferecida pela proposta do

Frameknow. Observou-se também que a participação direta dos membros das equipes na substituição de dados dos processos (correção dos dados incompletos ou faltosos) durante a execução das simulações e no “jogo dos dados” reais e simulada contribuiu para um comportamento mais maduro nas estimativas e no próprio trabalho do grupo, permitindo auto-reflexão da produtividade. Ambos os casos de estudos identificam causas e justificam-nas em relação aos períodos de trabalho mais e menos produtivo e também por estar relacionados a falta de um parâmetro interno da organização que identifique e aponte os objetivos ou metas de cada organização.

5. Conclusões e próximos passos

Pode-se concluir que o *Frameknow* é contribuição a área de engenharia de software e mais especificamente a gestão e processos e projetos de software. Uma vez que tendo um conjunto de funcionalidades específicas para o propósito mencionado, e estando disponíveis de serem adicionados a máquinas de processos e ambientes de projeto.

O tratamento de dados defeituosos e melhorias dos processos devem ser realizados no contexto da própria organização. O problema considerado, suas características, alternativas de soluções disponíveis como tecnologias, pessoal, infraestrutura, especificações em geral se realizam diante de entendimento e perfeita harmonia. Tudo isto integrado nos objetivos definidos ou planejados pela organização e da eficiente manutenção de ocorrências (fenômenos) diários decorrentes de sua própria natureza.

O ambiente de gestão de projetos adotado para subsidiar este protótipo (*dotProject*) serviu ao seu propósito, caracterizado pela integração ao contexto dos casos, conformando requisitos de representação direta das informações e controle dos processos, fácil operacionalização, de fácil entendimento e interpretação por todos os envolvidos..

Alguns dos futuros passos: a) ampliar e confirmar, em maior escala de resultados, as vantagens da proposta do *Frameknow* em diferentes situações comparando com propostas alternativas. b) incluir um *add-on* para manutenção de notação de processo no contexto da própria organização, baseado em ontologia. c) definir um gerenciador de versões dos processos, implementando recursos de monitoramento de versões, possibilitando tratamento de evoluções de conhecimento.

Referências bibliográficas

- Araújo, D. G. (2008) “*Frameknow*. Um *framework* para gestão de processos de software assistida por simulação baseado em conhecimento”. Tese de doutorado em andamento. Instituto Nacional de Pesquisas Espaciais. São Jose dos Campos-SP.
- Araújo, R. and Borges, M. (2002) “*Workflow* e gestão do conhecimento”. Caderno de Engenharia de Software. Universidade Federal do Rio de Janeiro. Rio de Janeiro-RJ.
- Armbrust, A. (2003) “Using empirical knowledge for software process simulation: A practice example”. Thesis. Fraunhofer IESE.
- Ioana, R. A.; Collofello, J. A. and Lakey, P. B. (1999) “Software process simulation for reliability management”. The Journal of Systems and Software, 46, 173-182, Nov.
- Kellner, M. I., Raffo, D. M. and Madachy, R. J. (1999) “Software process simulation modeling: Why? What? How?” The Journal of systems and software, 46, (2-3), 91-105 Apr.

- Reis, C. A. L. (2003) “Uma abordagem flexível para execução de processos de software evolutivos”. Doutorado. URGs. Porto Alegre-RS.
- Scacchi, W. (1999) “Experience with Software Process Simulation and Modeling”, ATRIUM Laboratory, University of Southern California . *Journal of Systems and Software*, Volume 46, Number 2, 15, pp. 183-192(10). Elsevier.
- Silva, F. A. D. (2001) “Um modelo de simulação de processos de software baseado em conhecimento para o ambiente PROSOFT”. Mestrado. UFRGS. Porto Alegre-RS.
- Silva, R. E. (2007) “Requisitos para integração de ferramentas de engenharia de software”. Relatório interno. Departamento de Ciências da Computação. UFMG. Belo Horizonte-MG.
- Souza, M. M. (2007) “Uma metodologia de predição estatística de projetos baseada em simulação”. Mestrado. Universidade Federal do Pernambuco. Recife-PE.
- Travassos, P. R. (2007) “Metodologia e ferramentas para a integração de simulação de processos com a gestão de projetos”. Doutorado. INPE. São José dos Campos-SP.
- Valente, F. F. R., Falbo, R. A. (2002) “Uso de gerência do conhecimento para apoiar a realização de estimativas”. *Centro Latinoamericano de Estudios en Informática*. Montevideo. Uruguay.
- Zelkowitz M. V. and Wallace, D. (1996) “Experimental models for software diagnosis, Natl. Inst of Stnds and Tech, NISTIR 5889, Sep.