

R-Cycle¹ : Um Molde para o Processo de Produção, Disponibilização e Evolução de Software

Luiz Mauricio F. Martins
J. Antão B. Moura
Álvaro F. C. Medeiros

Grupo de Sistemas de Software da UFPB/NIES/UFAL
CCT/DSC/COPIN

Av. Aprígio Veloso S/N, DSC, Sala Proj. Protem-CC/II
58109 - 970 Bodocongó Campina Grande - PB
Fone : (083) 3332128 Fax : (083) 3210781
E-Mail : luizm@cgsoft.softex.br
antao@cgsoft.softex.br
alvaro@dcc.ufal.br

Resumo

A prática demonstra que os modelos de processos de produção de software existentes não condizem com a realidade de mercado (software comercial), principalmente pela excessiva concentração em aspectos de desenvolvimento. Este artigo propõe um "molde" genérico mais realista e abrangente para a produção e disponibilização de software comercial, a partir do qual modelos específicos possam ser construídos e validados para casos particulares e realistas da indústria. O molde estende a abordagem tradicional dos modelos de processos, identificando fatores críticos para o sucesso de um software comercial, tais como marketing, comercialização e suporte técnico, que interferem diretamente nos prazos e orçamentos de projetos de software.

Abstract

Practice shows that the existing software production process models do not conform with reality of the software industry (commercial software) due to their great focus on development issues. This paper proposes a more realistic and generic template for modelling the production and delivery of commercial software upon which other models can be built and validated for specific and real industry cases. This template extends the traditional approach of process models identifying critical aspects for the success of a commercial software such as marketing, sales, distribution and technical support, which play an important role on software project budgets and schedules.

Palavras-Chave: Processo de Produção de Software, Modelos de Processos, Software Comercial, Engenharia de Software, Qualidade, Cooperação Universidade-Indústria

¹ Pesquisa financiada com recursos do CNPq/PNUD. Proc. Inst. N° 680071/94

Produzir software torna-se cada vez mais, uma atividade especializada, "eficiente" e competitiva. Esta afirmação é corroborada pela desistência da NEXT Inc em ser fornecedor de sistemas (hardware e software) para se especializar na oferta de software (o Next Step), pela observação da criação de novas empresas com vastos recursos (p ex., a Taligent, uma "joint-venture" da IBM e Apple para desenvolver um sistema operacional orientado a objetos) e pela acirrada concorrência do New Technology da Microsoft contra o Unix da Novell

A globalização do mercado de software, com os ganhos de escala, aumentou as pressões de concorrência. Assim, qualquer casa-de-software para competir com sucesso, deve ter eficiência no seu processo de produção. A modelagem deste processo ajuda a identificar aspectos importantes (merecedores portanto, de atenção continuada) de operação.

Observa-se também, um aumento da complexidade dos aplicativos e pacotes de software, na medida em que recursos mais sofisticados são oferecidos visando a satisfação e produtividade dos seus usuários [Mede94]. Consequentemente, o processo de produção adquiriu maior complexidade, podendo-se identificar diversas alternativas de execução das tarefas nele contidas. A modelagem do processo poderá destacar diretrizes para a seleção da alternativa mais adequada para uma determinada organização.

Definição - O "processo" de interesse aqui, como definido em [Hump89], consiste do conjunto de atividades, métodos e práticas utilizados na produção de software.

Para se detalhar melhor os objetivos e delimitar o alcance dos resultados a serem alcançados, redefine-se o foco do processo para incluir além da produção propriamente dita, a disponibilização (i.e., a entrega e suporte do produto final ao usuário) e a evolução (funcional, de desempenho, quanto à absorção de novas tecnologias, e à execução em outras plataformas / ambiente computacionais).

No que segue, fala-se então, de Processo de Produção, Disponibilização e Evolução (Processo PDE) de software.

No contexto brasileiro, as necessidades de modelagem do processo PDE são realçados pela iniciativa do Programa SoftEx 2000, que visa tornar o país um participante expressivo no mercado mundial de software. Este mercado movimentou 155 bilhões de dólares em 1992, segundo o International Data Corporation, e exige competência, dinamismo, preço e sobretudo qualidade para o sucesso de um produto de software. Ressalte-se que estas mesmas exigências são necessárias para garantir a auto sustentação da indústria nacional de software, no próprio mercado interno, frente à concorrência internacional já instalada.

De fato, este volume de negócios anunciado requer eficiência dos produtores nacionais, e pode ser almejado com a exata compreensão das atividades, métodos e práticas relacionadas com o processo de PDE no mercado. Modelos de produção são concebidos exatamente com o propósito de guiar Engenheiros de software e balisar as atividades de gerência quanto ao planejamento, organização e controle de todo o processo de produção [Ghez91].

Este trabalho se concentra na modelagem do processo PDE de software comercial, considerando suas particularidades em função da realidade verificada na prática de mercado.

A definição e adoção de um processo PDE de software por uma empresa apresentam dependência do perfil da organização produtora, do tipo de produto, do seu mercado alvo, das ferramentas e metodologias disponíveis e até mesmo do ambiente/sistema operacional.

Uma vez disponível, um modelo serve para disciplinar ações da equipe envolvida no processo e para auxiliar sua gerência, educando as estimativas de recursos.

Face às dependências apontadas, parece mais produtivo e abrangente se trabalhar com um "molde" (ou um "antemolde") mais genérico a partir do qual modelos mais específicos e adequados (consequentemente mais realistas e úteis, mas não necessariamente mais complexos) possam ser construídos e aplicados em função das dependências particulares de interesse.

Organização do artigo - Este artigo propõe um molde genérico como base para modelos PDE de software (comercial) que eliminam restrições de modelos existentes (discutidos na seção 2). O molde é proposto a partir de uma série de requisitos e objetivos básicos (seção 3) que definem sua especificação (apresentada na seção 4). Em seguida, o uso do molde é ilustrado na construção de um modelo PDE para software Unix de prateleira (cuja validação é comentada na seção 5). Conclusões e sugestões para continuação do trabalho são apresentadas na seção 6.

2. Modelos de Processos de Produção de Software

Quando se faz referência ao processo de desenvolvimento de software, normalmente se associa um modelo ao processo que se deve seguir para construir, liberar e evoluir o produto desde a concepção de uma idéia até a retirada do sistema de operação. Tal modelagem de processo é frequentemente citada na literatura como ciclo de vida do software [Ghez91].

Com a necessidade de se planejar e organizar melhor a construção de produtos de software, foram concebidos diversos modelos que tentaram disciplinar o ciclo de vida do software, definindo passos e atividades a serem fielmente seguidos. Cada um destes modelos possui ponto de vista próprio sobre o processo de produção do software. Dentre os modelos mais conhecidos e utilizados podemos citar o modelo em cascata, o modelo evolutivo, o modelo em espiral e o modelo transformacional, que formam basicamente os modelos tidos como clássicos.

O primeiro trabalho significativo, com vistas à necessidade de organização do processo de produção de software [Royc70], resultou no modelo "cascata" para o ciclo genérico de vida do produto. O modelo representa a produção de software como resultado da execução linear das seguintes atividades: estudo de viabilidade, análise e especificação de requisitos, projeto e especificação, codificação e teste de módulos, integração e testes de sistema, liberação e manutenção. O modelo cascata, apesar de ter introduzido disciplina no processo de produção de software, é mais indicado para software com requisitos bem conhecidos antecipadamente. A

rigidez do modelo dificulta acomodar mudanças de requisitos e gerenciais, ocorrências frequentes em projetos de software.

Os chamados modelos evolutivos atentaram para a incrementalidade que deve ser imposta ao processo de produção, baseando-se no princípio de que a primeira versão de um projeto deve ser considerada um protótipo descartável em função da imaturidade do conhecimento preciso dos seus requisitos [Broo75].

O modelo espiral [Boeh88], além de fornecer estrutura para aspectos técnicos, considera a análise de custos agregados durante o processo de produção como um fator de risco na gerência. Este metamodelo (pois acomoda outros modelos de produção) chama atenção para a continuidade cíclica da produção do software em evolução à linearidade do modelo cascata.

Ao se examinar a utilidade destes modelos, conclui-se que eles servem principalmente como instrumentos didáticos para explicar o processo de desenvolvimento, adotam fluxos de trabalhos genéricos [Hump89], e, que apresentam as seguintes restrições para serem aplicados à realidade de software no mercado:

- i) Não cobrem todo o ciclo de produção do software comercial (excluem por exemplo, aspectos críticos para o sucesso como empacotamento para distribuição, marketing e suporte técnico, que necessitam planejamento e procedimentos técnicos para realização);
- ii) Reforçam aspectos de desenvolvimento, desfocando a visão de atividades com maior fatia de custos (tais como suporte e comercialização);
- iii) Dificilmente assimilam mudanças estratégicas inesperadas no processo;
- iv) Não condizem com a realidade vivenciada no mercado, seja pelo excessivo formalismo, pela inadequação do modelo aos ambientes ou pela sua generalidade;
- v) São, na grande maioria, modelos orientados à tarefas, refletindo a familiaridade com outras engenharias que trabalham essencialmente com produtos manufaturados (software é uma atividade intelectual e multidisciplinar).

A situação atual da indústria de software como um todo é mais preocupante do que transparece pelas restrições acima na aplicação de modelos do processo PDE. A indústria parece operar sem modelo documentado ou compartilhado por membros da equipe.

Levantamentos preliminares realizados junto a casas-de-software nacionais, na tentativa de modelar o processo PDE de software comercial, evidenciaram que as atividades de produção e gerência de software deste segmento são realizadas empiricamente sem o apoio efetivo de modelos de processos. No acompanhamento de empresas emergentes junto ao programa SoftEX 2000, verificou-se a mesma carência.

A realidade no exterior não se distancia da vivenciada localmente. Dados coletados em estudo de campo realizado pelo Instituto de Engenharia de Software (SEI) da Universidade Carnegie Melon nos Estados Unidos [Kras92], mostram que a vasta maioria das organizações desenvolvedoras de software daquele país não possuem ou não usam modelos PDE. Uma das

exceções é a IBM, que se preocupa em construir modelos, que inclusive explicitam questões mais elaboradas como prototipagem e engenharia reversa, ao contrário dos modelos anteriores [IBM90].

O desinteresse da indústria pelo uso de modelos de processos PDE de software, pode ser atribuído parcialmente à dissociação de tais modelos da realidade de mercado.

Recentemente surgiram alguns trabalhos, ainda em fase de ajustes e validação, sugerindo mudanças na modelagem de processos, buscando apoiar efetivamente produtores de software diante do dinamismo e complexidade do processo de produção. O modelo COSMOS [Raym91] rompe com a arquitetura tradicional dos modelos, explicitando a necessidade de considerar aspectos gerenciais de infraestrutura e comunicação entre grupos no processo de produção em conjunto com as atividades técnicas tradicionais. Alguns princípios deste modelo são: coevolução de atividades, separação de domínios, melhoria contínua e prototipagem.

O modelo como um todo ainda se mostra muito incipiente, deixando em aberto a sua utilização como ferramenta de apoio para diversos tipos de ambientes e necessidades. Apesar de apresentar uma visão mais abrangente sobre a produção de software, não se pode afirmar que este venha a amenizar eficientemente a distância entre os modelos de processos existentes e a realidade de produção de software comercial. De uma maneira geral, são valorizadas as questões de gerência de comunicação de projeto e a infraestrutura necessária para a organização atingir seus objetivos de curto e longo prazo, ficando em aberto como o ciclo de produção deve ser conduzido.

Características de coevolução são exploradas no trabalho realizado internamente pela Fujitsu japonesa, que elaborou o Modelo de Processo de Desenvolvimento Concorrente (CDPM) [Aoya93]. Basicamente, o modelo prevê o desenvolvimento de múltiplas funções (módulos ou versões) concorrentemente, da especificação de requisitos até o sistema de testes, com alguns pontos de interferência. A utilização deste modelo em projetos com alto nível de comunicação com mais de um milhão de linhas de código fonte, acusou uma redução de 75% do tempo de desenvolvimento.

Algumas das restrições dos modelos clássicos para a produção de software comercial, identificadas acima, continuam no modelo CDPM. Isto se verifica principalmente porque, apesar do dito modelo utilizar frentes de trabalho simultâneas para reduzir o ciclo de produção, ele utiliza o modelo cascata como base para produção, enfocando apenas aspectos de desenvolvimento. Mesmo realizando uma reengenharia para adaptação do modelo cascata à realidade de produção de software concorrentemente, o CDPM continua não abordando o ciclo completo de produção, desconsiderando aspectos de mercado (p ex. marketing, suporte técnico e empacotamento) que permeiam a produção de software comercial.

3. Requisitos e Objetivos Básicos de um Modelo PDE

Molde vs. Modelo - Reiterando as restrições observadas nos modelos de processos existentes, a produção de software não pode ser considerada uma atividade linear, bem comportada e de acordo com algumas fases estáticas que visam simplesmente a construção de um produto.

De fato, tanto o processo de produção, quanto a construção de modelos efetivos no apoio do processo de produção de software são complexos. Isto porque tanto o processo quanto o modelo devem se ajustar dinamicamente a novas necessidades durante a sua execução, e contemplar a multidisciplinaridade do processo de produção, envolvendo atividades de gerência de processos e produtos.

Para possibilitar o aumento progressivo do conhecimento sobre modelagem de processos de produção de software, e com base em observações sobre o comportamento do mercado, foram desenvolvidos protótipos de modelos, denominados de "moldes".

A necessidade de um molde genérico surgiu à medida em que seria inadequado recomendar um único modelo representativo para qualquer tipo de software, empresa, equipe e segmento de mercado, como assim o fazem os modelos clássicos de processos. Propõe-se então, um molde representando genericamente a produção do software comercial, a partir do qual modelos específicos possam ser construídos e calibrados para casos particulares e realistas da indústria de software.

Requisitos - A definição de modelos de processos, clássicos ou novos, se baseia numa gama de requisitos que cada modelo em particular deseja atender. Assim como o modelo espiral se preocupa com custos, o modelo cascata com a execução linear de tarefas geridas por documentos, a modelagem do processo de produção de software comercial obrigatoriamente deve espelhar as realidades e práticas de mercado.

A partir do contacto com empresas nacionais (Infocon, Light Software, Sciencia e empresas emergentes vinculadas ao programa SoftEX 2000) e das limitações dos modelos de processos existentes, foram isolados alguns requisitos que demonstram as particularidades do software comercial em função das variantes de mercado. São eles :

- * *Abrangência*. São considerados todas as atividades envolvidas no ciclo de vida real do software, estendendo a abordagem tradicional. Aspectos como comercialização e sustentação de um produto são críticos para o sucesso de uma empresa de software.

- * *Separação de Domínios*. Decomposição do problema para facilitar a divisão de tarefas, permitindo a execução de estratégias de produção e de disponibilização, compatibilizando a complexidade do problema com os recursos disponíveis.

- * *Paralelismo*. Possibilidade de executar simultaneamente múltiplas funções de gerência de processos e de produtos. Prevê o paralelismo na implementação de rotinas (código, módulos) e até mesmo versões de um software em conjunto com outras atividades como marketing e testes, como meio de agilizar o ciclo do software.

- * *Feedback*. Capacidade que deve ter o modelo para retomar o processo de produção a fases ou atividades anteriores a qualquer momento, em virtude de problemas técnicos ou decisões estratégicas da empresa em função de marketing, concorrentes e tendências tecnológicas.

* *Integração*. Assimila as necessidades de cooperação entre as atividades multidisciplinares componentes da produção de software comercial : marketing, desenvolvimento, vendas, suporte técnico, administração e controle de qualidade.

* *Evolução*. Propriedade que garante a continuidade ciclica do produto no mercado. A utilização deste principio prevê o incremento de melhorias ou ajustes na última versão colocada no mercado.

* *Contexto mercadológico*. O modelo deve estar em completa sinergia com o mercado, destacando seus conceitos, valores e práticas, de forma a colocar a Engenharia de Software a serviço das necessidades mercadológicas.

* *Flexibilidade*. Conseguída com a agregação dos demais requisitos, dinamiza o processo de produção. A flexibilidade pode significar a assimilação de mudanças de requisitos, prazos, metas e orçamentos, pessoal, de ferramentas e plataformas de desenvolvimento.

A definição do molde na próxima seção é norteada pelo atendimento a estes requisitos e pela busca dos objetivos que seguem.

Objetivos - A adoção do molde e a construção de modelos nele baseados devem atender especificamente às necessidades dos diversos segmentos de mercado, fornecendo os seguintes resultados :

- ✓ Tomar a Engenharia Software mais Realista e Abrangente
- ✓ Fornecer uma visualização e compreensão de todo o processo de produção
- ✓ Chamar a atenção para alguns aspectos críticos para o sucesso no processo PDE de software
- ✓ Sugerir diretrizes e possibilidades de automação nas diversas fases e atividades do processo
- ✓ Apoiar a gerência do processo com métricas aplicáveis a diversas fases (não apenas em desenvolvimento)
- ✓ Sugerir procedimentos operacionais práticos e realistas para maior competitividade da indústria nacional

O molde a ser proposto e ao atingir os objetivos acima, alinha-se com recomendações recentes [Pott93] para que a Engenharia de Software se aproxime mais da realidade de mercado e apoie mais eficazmente a indústria.

4. Um Molde Mais Realista

A construção de um molde representativo para o processo de produção de software comercial, fundamentado nos principios descritos, se iniciou com a caracterização dos ele-

mentos componentes do ciclo de vida do software comercial e seu comportamento em relação ao tempo.

Segundo o molde (ilustrado na figura 1), o processo de produção de software comercial pode ser organizado basicamente em torno de três componentes integrados: fases, atividades monofásicas e atividades polifásicas.

Assim sendo, o molde verifica que num nível mais alto de abstração encontram-se as **fases**, que identificam os principais estágios distintos no processo de produção de software, independentemente do segmento de mercado. Cada uma das fases possui atividades correlatas denominadas **atividades monofásicas** que operacionalizam as fases de acordo com a realidade de cada segmento de mercado. Por fim, as **atividades polifásicas** fornecem a sustentação necessária para o sucesso do produto durante todo o ciclo de vida real do software.

O molde identifica então, quatro fases (representadas na figura 1 pelas caixas) que caracterizam genericamente o ciclo de vida de um software comercial, descritas a seguir com suas respectivas atividades monofásicas (visualizadas na figura na parte frontal das caixas) :

- ❑ **Fase 1 : Concepção** - Se concentra no processo de identificação da informação a ser processada e é disparada seja pela descoberta de oportunidades no mercado, solicitações de clientes, contratos com parceiros ou evolução planejada de um software já disponível. Nesta fase se discute amplamente *o quê* realmente se quer conceber em termos de funcionalidade, interfaces, performances, critérios de validação e restrições de projeto, para ficarem estabelecidos os parâmetros de um projeto bem sucedido para a gerência, desenvolvedores e futuros usuários.
- **Atividades Monofásicas (da fase 1)** : são realizados estudos para viabilização e planejamento do projeto, para que de uma maneira geral sejam estimados os prazos e custos, e captados os recursos necessários (pessoal, equipamento, capital, etc). A atividade monofásica de especificação conclui esta etapa, apresentando a especificação completa do software juntamente com o manual do usuário. Neste momento do ciclo de vida, a equipe encarregada tem a missão de definir mais especificamente *como* o software será projetado, implementado e com seus componente
- ❑ **Fase 2 : Desenvolvimento** - tem como objetivo transformar a especificação em um sistemas testados [Pres87]
- **Atividades Monofásicas (da fase 2)** : a Engenharia de Software concentra atualmente grande parcela dos seus trabalhos na automação das atividades monofásicas de projeto (arquitetural, procedural, estrutura de dados e interface), codificação, integração de programas e realização de testes individuais e de integração.
- ❑ **Fase 3 : Preparação** - esta fase é incluída a partir da necessidade de transformar o software num produto destinado ao mercado após ser liberado pela equipe de desenvolvimento. Esta fase envolve aspectos de marketing (e principalmente) técnicos.

• **Atividades Monofásicas (da fase 3)** : Antes do produto ser comercializado, os produtores de software costumam submeter o software a apreciação dentro da própria empresa e no mercado consumidor em potencial, com o produto acabado. Estes atividades monofásicas, denominadas respectivamente de testes alfa e beta, são consideradas como pontos de decisão (*milestones*) no ciclo de vida do software comercial, pois podem impactar na retomada de alguma atividade anterior para melhorar a aceitação do produto. Aprovado o produto, a atividade de empacotamento é responsável pela execução de procedimentos que controlam a geração do software para utilização no mercado (versões, mídias, documentação, tipo de licença), de forma a garantir a instalação adequada na máquina do usuário e proteger o software contra utilização indevida. O empacotamento, apesar de parecer ser restrito à tarefas administrativas, requer implementação de esquemas e rotinas, utilização de ferramentas e controle de qualidade.

- **Fase 4 : Disponibilização** - também referenciada na literatura como manutenção, objetiva suportar tecnicamente os produtos vendidos e gerenciar o processo de mudanças, fruto de erros detectados e alterações de requisitos de usuários ou do próprio ambiente. A disponibilização responde pela continuidade do ciclo para um produto de software existente, reiniciando o ciclo de produção com novos requisitos

• **Atividades monofásicas (da fase 4)** : dizem respeito às estratégias de vendas e distribuição do software, suporte técnico aos usuários e manutenção (corretiva, evolutiva, adaptativa ou perfeita) do software ao longo do tempo .

Atividades Polifásicas - De acordo com o molde proposto ilustrado na figura 1, são detectadas atividades que ocorrem em mais de uma fase ou durante todo o ciclo de vida do software, demonstrando serem de extrema importância para a gestão, controle e acompanhamento do processo PDE. Estas atividades, denominadas atividades polifásicas ou de acompanhamento, possuem normalmente estrutura de funcionamento e procedimentos próprios. As principais atividades polifásicas identificadas pelo molde são: *Investimento, Documentação, Marketing e Controle de Qualidade*.

As atividades polifásicas são extremamente importantes para a representatividade do molde, uma vez que podem alterar o fluxo convencional do ciclo de vida em função da realidade de mercado. Ao se considerar, por exemplo, o movimento de concorrentes do mercado através do marketing e a disponibilidade de investimentos da empresa, um produtor pode decidir retornar à fases anteriores para redesenho do projeto ou inclusão de novas características

Investimentos - Esta atividade pode ser balizada inicialmente por alguns dos estudos preditivos de custos reportados na bibliografia [Boeh81] , [Dema89]. Entretanto estes estudos tipicamente (e mais uma vez) se concentram em apenas algumas das fases. Aqui, o molde ressalta a necessidade de se endereçarem questões importantes de comercialização, suporte técnico e empacotamento de produtos, responsáveis por grande fatia dos custos totais do processo PDE. Esta consideração pode a princípio ser feita por simples contabilidade e relatórios financeiros mensais que instrumentalizem a gerência de todo o processo PDE para garantir a disponibilidade de recursos em todas as fases. A garantia de distribuição de investimentos por todas as fases devidamente dosadas com a qualidade do produto trará melhores chances de sucesso no mercado.

Documentação - De uma forma geral, visa :

- registrar as atividades técnicas de concepção e desenvolvimento do software
- orientar o usuário (p.ex., manual de instalação, manual do usuário)
- controlar a operação da empresa (p.ex., contratos de negócios, resultados de investimentos, registro do software, direitos autorais)
- controlar a qualidade do software (p.ex., inspeções, resultados de testes)
- apoiar a área comercial (p.ex., descrição de produtos, perguntas & respostas de vendas, roteiro de apresentação do produto, resumo executivo, comparativo com concorrentes)
- amparar o suporte no fornecimento de notas técnicas para usuários, *workarounds*, dentre outros.

A obtenção de registros das atividades do processo de produção nos seus diversos estágios e disciplinas, forma a base para a implantação de melhorias e auxilia a gerência na estimativa de novos projetos, a partir do estabelecimento histórico do ambiente.

Marketing - Esta atividade polifásica não se restringe às atividades de divulgação do produto para o seu público alvo. O marketing é também útil na detecção de oportunidades para o desenvolvimento e/ou evolução de produtos e levantamento da viabilidade de mercado. Talvez mais importante do que saber como fazer o produto é saber se alguém quer consumi-lo e se esta demanda justifica os investimentos a serem realizados para a sua produção. Após viabilizar a produção e estabelecer metas para o produto que será desenvolvido, o marketing deve preparar o mercado, gerando expectativa de consumo. Uma vez que o produto esteja no mercado, o marketing deve monitorar a sua aceitação e seus possíveis concorrentes, determinando as estratégias para sua sustentação e evolução.

Controle de Qualidade - O molde permite a adoção direta das normas ISO 9001 / 9000-3 (elas próprias de natureza polifásica) para controle total da qualidade do processo PDE completo, já que o molde inclui interfaces com a equipe de produção e com os usuários (através dos testes beta, vendas, distribuição, suporte técnico e marketing).

Vale ainda ressaltar que a inclusão da noção de tempo impõe dinamismo ao molde, possibilitando a execução de fases concorrentemente e a evolução de produtos para novas versões.

5. Ilustração da Aplicação do Molde

A viabilidade do uso do molde descrito como base para construção de modelos realistas (e mais completos) de processo PDE de software foi estudada no contexto de uma casa de software de Campina Grande que produz software de prateleira para o segmento de mercado Unix. Como pode ser visto na dissertação de mestrado na área de Engenharia de Software na UFPb [Mart93], este trabalho buscou apresentar uma instanciação do uso do molde na construção de um modelo para um segmento de mercado específico.

O estudo focalizou principalmente as fases de preparação e disponibilização, consideradas vitais para o sucesso do produto [Kras92], desde que já há concentração satisfatória em concepção e desenvolvimento pelos modelos clássicos da Engenharia de Software.

Resultados preliminares ofereceram benefícios para a academia e a indústria. Com o modelo resultante, a empresa produtora pode compreender e ordenar melhor as atividades do seu negócio, identificando explicitamente as áreas merecedoras de atenção em função de suas deficiências e investimentos realizados. Métodos e práticas para cada atividade (mono ou polifásica) foram selecionados criteriosamente, documentados e calibrados para necessidades e peculiaridades da empresa, dos seus produtos, equipe e mercado.

É interessante notar que fases aparentemente não-técnicas (como preparação e disponibilização) exigiram, nesta aplicação do molde, métodos e práticas eminentemente técnicas para maior eficiência (p. ex., rotinas auxiliares para empacotamento do produto em termos de instalação, segurança; para amenizar suporte remoto em termos de configuradores, descrição de dispositivos e "work-arounds" para bugs residuais, e, para apoio na comercialização como desenvolvimento de versões para demonstração, homologação e cópias-"full" foram especificadas, documentadas, implementadas e incorporadas ao produto).

A UFPb, por sua vez, obteve um retorno do mercado sobre as diretrizes de produção, disponibilização e evolução de produtos Unix, e pretende atualizar seu curso de pós-graduação em Informática a partir da estrutura fornecida pelo molde.

Enfim, cada ambiente de produção de software instancia o molde de acordo com sua realidade, o qual servirá de guia para a construção do seu próprio modelo que represente suas peculiaridades eficazmente.

NIES / UFAL

Este molde encontra-se, atualmente, em fase de instanciação para softwares educacionais no ambiente Windows, por pesquisadores do Núcleo de Informática Educativa no Ensino Superior (NIES) da Universidade Federal de Alagoas. Este experimento é parte das atividades de pesquisa do Projeto R-Cycle financiado pelo CNPq/PNUD.

6. Conclusão

Este artigo apresentou um "molde" a partir do qual modelos adequados à realidade de processos de produção de software adotados na prática podem ser elaborados.

Sua aplicação por partes de novas empresas de software ligadas ao programa Softex2000 demonstra sua viabilidade. O fato do molde integrar atividades críticas do processo de produção de software comercial, tais como: marketing, empacotamento, testes alfa e beta, controle de qualidade e suporte pré e pós-venda, faz com que o produtor de software consiga ter uma visão mais geral de todo o processo e não se limite aos aspectos apenas de codificação.

O molde proposto visa contribuir para que a engenharia de software se torne mais realista e abrangente. Mais realista na medida em que retrata a realidade de produção de software, considerando as práticas de mercado e se constituindo num instrumento efetivo para o planejamento e controle do processo de produção de software; mais abrangente, por estender a abordagem tradicional centrada apenas no desenvolvimento de produtos, destacando a importância das fases de preparação e disponibilização do software para o mercado, assim como a influência das atividades polifásicas no processo.

Os levantamentos realizados com o produtor local para instanciação do molde para software Unix de prateleira demonstram que existe uma extrema necessidade de compreensão e ordenação do negócio. A produção de software é uma atividade multidisciplinar, que requer um relacionamento estreito com outras áreas de estudo, tais como: controle de qualidade, administração, economia, direito, marketing, vendas, comércio exterior, merchandising. Por isto e pelo nível de especialização em que atingiu a produção de software, tanto o processo de produção quanto os modelos que os representam são complexos. Entretanto, os resultados conseguidos foram animadores para este produtor, que pode visualizar melhor a complexidade do seu processo de produção com a utilização de um molde aplicado ao seu segmento de mercado.

A visualização e compreensão do ciclo completo de produção de software comercial pode contribuir efetivamente para empresas produtoras de software. Neste sentido, acredita-se que este trabalho seja uma contribuição efetiva para o programa SoftEX 2000, à medida em que baliza as empresas emergentes sobre a complexidade de se produzir software para o exigente mercado internacional.

Trabalhos Futuros - Esforços devem ser concentrados na construção e validação de modelos apropriados para perfis específicos de produtores de software (empresas de terceirização, de mercado livre, de desenvolvimento interno), tipos de produtos (prateleira, sob-encomenda, embutido) e segmentos de mercado (Unix, Windows, OS/2). Os esforços devem ser realizados a partir de estreita cooperação de grupos acadêmicos interessados em Engenharia de Software e a indústria, como ilustra a figura 2.

Nossa recomendação de execução e distribuição de trabalhos futuros nesta linha de pesquisa é feita em função da vocação, *expertise* e experiência, como ilustra a figura 2. A partir de experiências relatadas pela indústria, os grupos acadêmicos envolvidos no projeto formalizam e organizam os diversos modelos (possivelmente com o engajamento de alunos de mestrado), bem como especificam e desenvolvem ferramentas de automação de alguma(s) atividade(s). Os modelos e ferramentas construídos são validados em campo pela indústria. Uma vez validados, os modelos poderão ser adotados pela academia na atualização do ensino e/ou calibrados pela indústria para uso interno.



Figura 2 - Cooperação Academia-Indústria

Baseando-se na descrição do molde proposto, o processo de produção de software incorpora algumas atividades (monofásicas e polifásicas) ainda não devidamente exploradas pela engenharia de software e que merecem destaque pela sua importância no processo (empacotamento, suporte, testes alfa e beta, manutenção etc). Esta noção abre a oportunidade para o detalhamento de rotinas e procedimentos, bem como a construção de ferramentas que automatizem as diversas atividades previstas, sejam estas mono ou polifásicas.

Com relação ao detalhamento de atividades monofásicas, pode-se obter por exemplo, uma especificação mais detalhada de um sistema de automatização do suporte técnico integrado com as outras atividades do molde. Talvez o suporte seja uma das mais importantes atividades no ciclo de vida do software, pela necessidade de sustentação técnica (pré e pós-venda) e evolução de produtos, e atualmente é carente de ferramental e bibliografias específicas. A especificação e implementação de um trabalho deste porte seria uma contribuição importante da academia para o setor produtivo.

A integração das atividades polifásicas ao processo de produção deve ser mais detalhada pelo valor estratégico que estas possuem. Existem trabalhos sobre custos, documentação, qualidade e marketing direcionados ao processo de produção de software, os quais poderiam ser mais interessantes caso estendessem a sua abordagem sobre todo o processo de produção e disponibilização de software. Por exemplo, um modelo de custos realista evidentemente teria que incluir aspectos de preparação e disponibilização, não ficando restrito à estimativas sobre o desenvolvimento de produtos.

Agradecimentos - Os autores agradecem o suporte financeiro do CNPq e da CAPES a este trabalho.

7. Referências

- [Aoya93] Aoyama, Mikio. *Concurrent Development Process Model*. IEEE Software, July 1993, pp. 46-55.
- [Boeh81] Boehm, Barry. *Software Engineering Economics*, USA, Englewood Cliffs, Prentice Hall, 1981.
- [Dema89] DeMarco, Tom. *Controle de Projetos de Software*, Rio de Janeiro, Campus, 1989.
- [Boeh88] Boehm, B.W. *A Spiral Model of Software Development and Enhancement*. IEEE Computer, 21(5): 61-72, May 1988.
- [Broo75] Brooks, F.P. *The Mythical Man-Months: Essays on Software Engineering*, Addison-Wesley, Mass., 1975.
- [Ghez91] Ghezzi, C. et alli. *Fundamentals of Software Engineering*. Prentice-Hall, Englewoods Cliffs, NJ, 1991.
- [Hump89] Humphrey, W.S. *Managing the Software Process*. Addison-Wesley, Mass., 1989.
- [IBM90] Software-First Life Cycle Final Definition Document. USAF ESD Contract F19628-88-D-0032, CDRL # 1240 from IBM FSD, Jan, 1990.
- [Kras92] Krasner, Herb et alli. *Lessons Learned from a Software Process Modeling System*. Communications of the ACM, 35,9, September 1992, pp 91-100.
- [Mart93] Martins, L.M. *Um Molde para o Processo de Produção e Disponibilização de Software Comercial. Ilustrações com Software Unix de Prateleira*. Dissertação de Mestrado, Universidade Federal da Paraíba, December 1993.
- [Mede94] Medeiros, A. F. C.; SAUVÉ, J. P.; MOURA, J. A. B.; NICOLETI, P. S. - *Aumentando a Produtividade e Qualidade em Sistemas Abertos: Guia Avançado para Ambientes UNIX* - Makron Books, 1994;
- [Pott93] Potts, C. *Software Engineering Research Revisited*. IEEE Software, September 1993, pp 19-28.
- [Pres87] Pressman, R.H. *Software Engineering: A Practitioner's Approach*, USA, McGraw-Hill, 1987.
- [Raym91] Yeh, Raymond T. *A Commonsense Management Model*. IEEE Software, November 1991, pp. 23-33.
- [Royc70] Royce, W.W. *Managing the Development of Large Software Systems: Concepts and Techniques*. Proceedings Wescon, August 1970.