

Uma Proposta de um Ambiente de Implementação de Processo de Software

SANDRO RONALDO BEZERRA OLIVEIRA ^{1,2}
ALEXANDRE MARCOS LINS DE VASCONCELOS ²
ANA CRISTINA ROUILLER ³

¹Centro de Ciências Exatas e Tecnologia – Universidade da Amazônia (UNAMA)
Av. Alcindo Cacela, 287, 66060-902, Belém – PA – Brasil
Fone: (+55 91) 210-3000, Fax: (+55 91) 225-3909

²Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851, 50732-970, Recife – PE – Brasil
Fone / Fax: (+55 81) 2126-8430

³Departamento de Ciência da Computação – Universidade Federal de Lavras (UFLA)
Caixa Postal 3010, 37200-000, Lavras – MG – Brasil
Fone / Fax: (+55 35) 3829-1123

(srbo, amlv)@cin.ufpe.br, acr@comp.ufla.br

Resumo. Ambientes de Desenvolvimento de Software Centrados no Processo possuem como uma de suas finalidades permitir que as etapas do ciclo de vida de processo de software (definição, simulação, execução e avaliação) possam ser automatizadas. Este artigo apresenta a arquitetura de implementação dos serviços contemplados para uma abordagem de automação e implementação de processos de software usando como base a especificação de projeto de um ambiente. Este processo deve ser instanciado a partir das características e propriedades que definem uma organização ou um domínio de um projeto específico. Este trabalho é uma pesquisa do programa de Doutorado do CIN/UFPE.

Palavras-chave: Processo de Software, Automação e Gerenciamento do Processo de Software, Qualidade de Processo de Software, Ambientes de Desenvolvimento de Software.

1. Introdução

Apesar dos inúmeros avanços na Engenharia de Software, muito ainda é discutido acerca da baixa qualidade e produtividade da indústria mundial de software, refletindo-se na insatisfação dos seus usuários e em prejuízos financeiros de enormes proporções. Os computadores estão rapidamente tornando-se componentes comuns do dia-a-dia das pessoas que, por sua vez, apontam necessidades com requisitos de complexidade cada vez maiores.

Com o objetivo de solucionar estes problemas, várias tecnologias vêm sendo experimentadas dentro do contexto de apoiar o ciclo de vida do software. Um dos esforços mais significativos corresponde à definição de metodologias voltadas a disciplinar o processo de desenvolvimento através do estabelecimento de etapas bem definidas,

proporcionando, desta forma, um mecanismo de controle para o processo. Durante estas etapas, segundo [Reis 2000], os desenvolvedores envolvem-se na tarefa de construir modelos que, descritos em notações, atuam como mecanismos de compreensão dos requisitos e comunicação com outros profissionais, e sistematicamente são transformados em uma representação computável (software).

O surgimento da tecnologia CASE (*Computer Aided Software Engineering*) - Engenharia de Software Auxiliada por Computador, exerceu um enorme impacto sobre a área. As ferramentas CASE proporcionam uma sólida estrutura às metodologias e métodos de desenvolvimento de software. Os ambientes integrados de desenvolvimento de software, ou simplesmente ambientes de desenvolvimento de software (ADSs)

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

O apoio à gerência do processo de desenvolvimento de software corresponde a um dos requisitos que são exigidos nos ADSs atuais. Técnicas e ferramentas são integradas para permitir o controle de cada uma das etapas que são necessárias para desenvolver um produto de software. Assim sendo, os ambientes de desenvolvimento de software orientados ao processo [Gimenes 1994][Reis 2000] surgiram com o objetivo de proporcionar uma estrutura computacional que gerencie o intercâmbio de informações entre os desenvolvedores, controlando as atividades realizadas, envolvendo assim os recursos consumidos, os prazos determinados, e as datas de início e término de cada atividade.

Uma das evoluções mais importantes no estudo da qualidade está em notar que a qualidade do produto é algo bom, mas que qualidade do processo de produção é ainda mais importante [Junior 1997]. No caso do software, por exemplo, pode-se dizer mais sobre a qualidade observando como o software foi desenvolvido ao invés de analisar apenas o produto final. Afinal, não se consegue ter certeza da manutenibilidade ou fidedignidade apenas usando a aplicação desenvolvida. Os estudos sobre qualidade mais recentes são, na sua maioria, voltados para o melhoramento do processo de desenvolvimento de software. Não é que a qualidade do produto não seja importante, o fato é que, ao garantir a qualidade do processo, já se está dando um grande passo para garantir também a qualidade do produto [Junior 1997].

Deste modo, surgiram importantes mecanismos de certificação da qualidade de software que são baseados na maturidade atingida pelas organizações de desenvolvimento de software na condução dos seus processos. Assim, padrões internacionais de qualidade de software como o CMM [Paulk 1993] do CMU-SEI e o ISO-15504 (SPICE) [ISO 1998] vêm sendo amplamente utilizados para avaliar a qualidade de software a partir do processo adotado para o seu desenvolvimento. A tecnologia de processo de software é uma importante aliada na busca da melhoria da qualidade do software produzido, pois busca aplicar um apoio automatizado à gerência do processo de desenvolvimento de software.

Por outro lado, para apoiar a modelagem e a execução de processo, têm sido propostos ambientes de desenvolvimento de software centrados no processo, os quais englobam, além das ferramentas de apoio ao desenvolvedor, ferramentas que permitem a modelagem do processo de software e a execução do mesmo. Desta forma, o ambiente “conhece” o processo a ser seguido e pode, assim, orientar os desenvolvedores na execução de suas tarefas, além de executar automaticamente tarefas repetitivas [Reis 2000].

Muito já se discutiu sobre as propriedades deste tipo de tecnologia, no entanto, percebe-se ao longo da execução do processo, a partir destes ambientes de desenvolvimento, que sua implementação nem sempre satisfaz as necessidades das organizações ou dos projetos desenvolvidos por estas. Isto se deve ao fato de que os responsáveis pela definição do processo não dispõem de um guia contendo as suas reais necessidades de execução, indicando as melhores práticas a serem instanciadas a partir de um processo padrão.

Neste contexto, é possível encontrar na comunidade inúmeros ambientes de desenvolvimento de software centrados no processo, onde nem todas as atividades que perfazem o ciclo de vida de um processo de software (Definição, Simulação, Execução e Avaliação) são atendidas em sua completude, como o PROSOFT [Reis 2003], Odissey Share [Werner 2003], Provision [Proforma 2000], Adele-Tempo [Belkhatir 1994], SPADE [Bandinelli. 1994], ProcessWeaver [Christie 1995], EPOS [Nguyen 1996].

No entanto o que detectou-se, a partir da análise das características de cada um desses ambientes, é que nenhum desses se preocupa com alguns pontos-chave de discussão atual: definição de processo baseada na detecção de aspectos que caracterizem uma classe de um sistema de software específico e propriedades que definem a estrutura de uma organização; uso de lições aprendidas ao longo de definições de processos, possibilitando desta forma que o ambiente sugira ao usuário atributos de composição do processo; adequação de atividades que especifiquem melhoria contínua no processo de software; reutilização de processos de acordo com o nível de sua definição (a ser discutido nas seções a seguir); plena automação das atividades que definem um ciclo de vida de processo de software,

seja usando ferramentas já existentes na comunidade, seja pela adaptação destas.

Surge, então, a necessidade da definição de um ambiente para a implementação de processo de software. Este ambiente tende a possibilitar a especificação dos processos de acordo com o domínio do projeto específico e das características da organização; a instanciação do processo de software para propriedades dos projetos; sua simulação a partir dos parâmetros de configuração (prazos, pressões internas e externas à organização, custos, recursos, etc.); uma execução (automação) mais próxima do que se espera para um processo organizacional; e uma avaliação a partir da coleta de métricas desta execução. O projeto deste ambiente de implementação de processos de software encontra-se descrito neste artigo.

Além desta seção introdutória, o artigo apresenta outras duas seções. Na seção 2 serão abordadas as propriedades que compõem um ambiente de implementação de processo de software. Na seção 3 serão apresentadas as considerações finais deste artigo.

2. Especificação de um Ambiente de Implementação de Processo de Software

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

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

- Especificar um meta-modelo de processo de software a fim de definir uma terminologia única entre os vários modelos de qualidade de processo de software existentes;
- Apoiar a definição de um processo de software para organização;
- Permitir a modelagem e instanciação deste

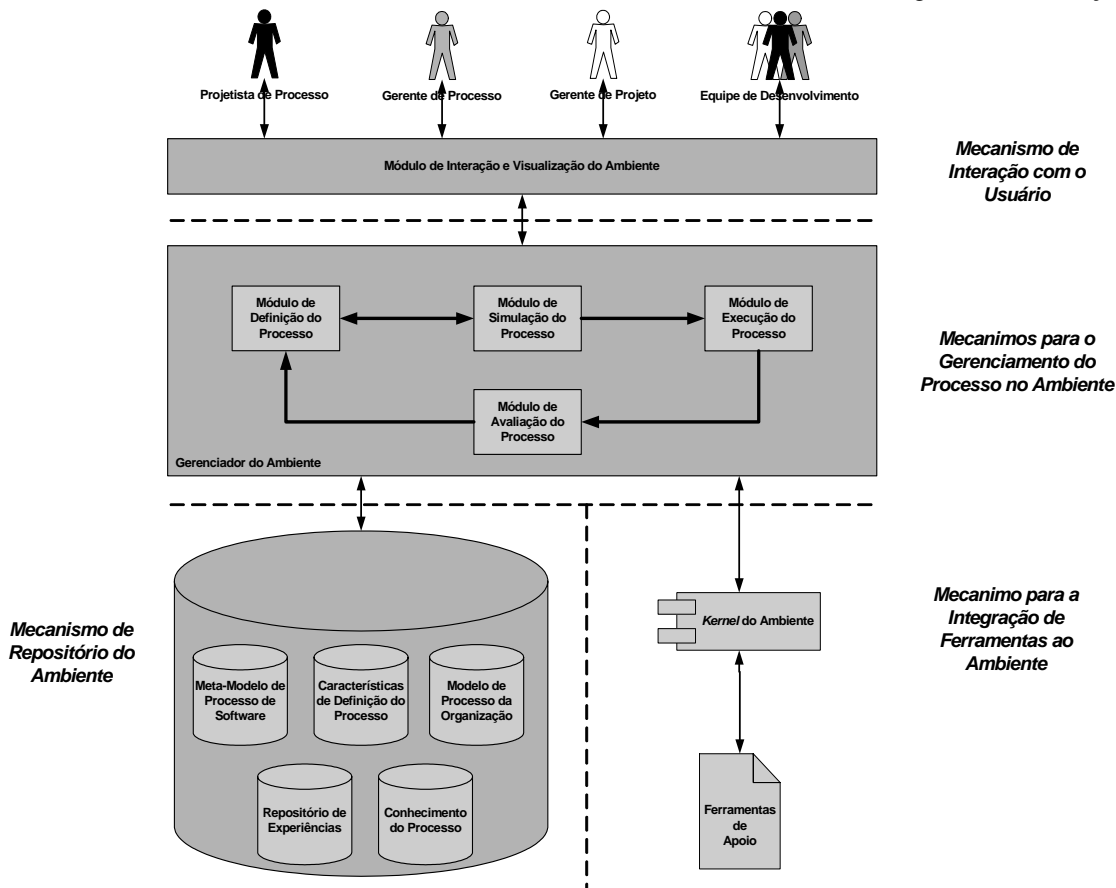


Figura 1: Arquitetura do Ambiente de Implementação de Processo de Software

processo;

- Permitir a simulação do processo a partir das características instanciadas para um projeto específico;
- Dar apoio à execução do processo de software;
- Possibilitar a avaliação dos critérios do processo de software;
- Apoiar a melhoria contínua do processo de software, através da realimentação e coleta de experiências aprendidas.

Vale ressaltar que todos os objetivos listados acima foram adaptados a partir da estrutura que compõe o meta-processo de software descrito em [Reis 2000], das características propostas para a implementação de um processo de software [Balduino 2002] e do ciclo de vida para melhoria contínua de processo definido pelo Modelo IDEAL [Mcfeeley 1996]. Para alcançar estes objetivos o ambiente foi concebido para adotar a arquitetura apresentada na Figura 1.

Pode-se notar que a arquitetura contempla quatro tipos de usuários para interação com o Ambiente:

- **Projetista do Processo:** responsável pela definição do processo e coleta de experiência sobre a execução de projetos. Este tipo de usuário interage com o ambiente recebendo orientações e identificando melhorias para processos existentes ou em concepção;
- **Gerente de Processo:** acompanha a simulação e a avaliação do processo a fim de prover conhecimentos formal e informal para possibilitar o reuso e a melhoria contínua dos processos;
- **Gerente de Projetos:** este usuário atua nas fases de instanciamento do processo para um projeto específico, acompanhando a execução do processo e a sua avaliação para posterior coleta de experiências;
- **Equipe de Desenvolvimento:** agrupa todos os perfis relacionados à execução de um projeto de software (Gerentes, Analistas, Engenheiros de Software, Arquitetos, etc.).

Os componentes definidos na arquitetura do ambiente encontram-se resumidamente descritos nas sub-seções a seguir.

2.1. Mecanismo de Interação com o Usuário

Nesse mecanismo o foco está em prover, aos usuários envolvidos com os serviços do ambiente, diferentes visões da mesma informação sendo definida e especificada. Este mecanismo provê interação para diferentes usuários do ambiente (por exemplo, agendas para equipe de desenvolvimento e facilidade de visualização de processo para projetistas/gerentes), ou seja, trabalha com as características da usabilidade do processo de software no ambiente.

2.2. Mecanismo para o Gerenciamento do Processo no Ambiente

Este mecanismo possui a responsabilidade de prover os serviços (definição, simulação, execução e avaliação do processo de software) especificados ao ambiente de forma automatizada, ou seja, possibilitar que os usuários do ambiente executem suas funções tendo como referencial um guia.

Ele é composto por alguns módulos, onde alguns destes (simulação e execução) possuem suas funções executadas por ferramentas de apoio integradas ao ambiente, tendo, desta forma, a finalidade de gerenciar apenas o uso destas ferramentas e em alguns casos agregar novos serviços que contemplem as funções do ambiente:

- **Módulo de Definição do Processo:** responsável pela definição do processo de software em níveis de caracterização (Processo Padrão, Processo Especializado e Processo Instanciado), melhor explicado a seguir; pela modelagem a partir de uma linguagem de representação diagramática no ambiente permitindo a visualização do processo definido e o relacionamento dos seus componentes; pela instanciamento através de regras que especifiquem o domínio de atuação do processo de software para um projeto específico permitindo que este processo possa ser executado por uma máquina de processos; e pela execução das funções de melhoria contínua e de reuso do processo de software através da coleta e filtragem de experiências obtidas;
- **Módulo de Simulação do Processo:** permite o refinamento dos modelos de processos de software, antevendo resultados da realização do projeto a partir da condução de experimentos baseados em computador a

fim de descrever, explicar e prever o comportamento de um sistema real;

- **Módulo de Execução do Processo:** coordena as atividades do processo em execução (através da invocação de ferramentas de projeto), podendo executar processos incompletos e permitindo alteração do processo durante a execução;
- **Módulo de Avaliação do Processo:** provê uma avaliação da execução do processo de software, descrevendo o desempenho dos componentes definidos para a estrutura do processo. Ocorre simultaneamente à execução do modelo de processo e as informações adquiridas são utilizadas no módulo de Definição do Processo a fim de coletar experiências para que estas possam compor um processo existente ou armazenar o conhecimento adquirido para novos processos.

Na estrutura definida na Figura 3 (notação representada na Figura 2), adaptada do modelo definido por [Rocha 2001], inicialmente observa-se o Meta-modelo de processo de software, composto de componentes e dos relacionamentos entre esses que são oriundos do mapeamento de algumas normas e modelos de qualidade para processo de software (CMMI [Chrissis 2003], SPICE – ISO 15504 [ISO 1998], ISO 9000-3 [ISO 1997]). O objetivo deste meta-modelo é determinar uma terminologia única para a definição de processos de software no Ambiente.

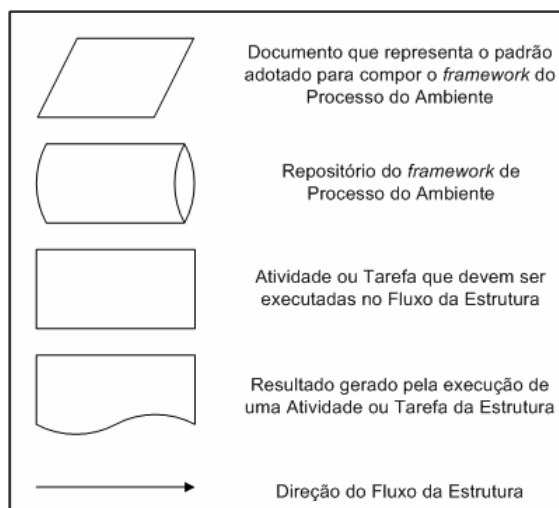


Figura 2: Primitivas da Estrutura da Figura 3

Por sua vez, a definição de um processo padrão estabelece uma estrutura comum a ser utilizada pela organização nos seus projetos de software e constitui a base para a definição de todos os seus processos. Dessa forma, estabelece-se um processo básico que servirá como ponto de partida para a posterior definição dos processos de software adequados às diferentes características de cada projeto, permitindo economia de tempo e esforço na definição de novos processos.

Tendo em vista que tipos de software diferentes possuem características distintas e requerem diferentes abordagens de desenvolvimento, o processo de software padrão da organização deverá ser adaptado (especializado) considerando-se as características relacionadas ao tipo de software (por exemplo, sistemas de informação) e ao paradigma de desenvolvimento utilizado (por exemplo, orientação a objetos). Assim, durante a etapa de especialização do processo padrão, atividades poderão ser adicionadas ou modificadas, de acordo com o contexto para qual se está realizando a especialização.

A instanciação para projetos específicos consiste na adaptação de um processo especializado a um projeto, considerando-se as suas peculiaridades. Nesta etapa, são definidos o modelo de ciclo de vida, os métodos e as ferramentas que serão utilizadas no projeto, os recursos humanos e suas responsabilidades ao longo do processo e os artefatos (produtos) consumidos e gerados.

2.3. Mecanismo de Repositório do Ambiente

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

2.4. Mecanismo para Integração de Ferramentas ao Ambiente

Este mecanismo provê a integração do ambiente com outras ferramentas tanto de apoio ao processo de software, quanto à execução do projeto de software, possibilitando desta forma a automação de atividades definidas no processo de software e a execução dos serviços providos pelos módulos de simulação e execução.

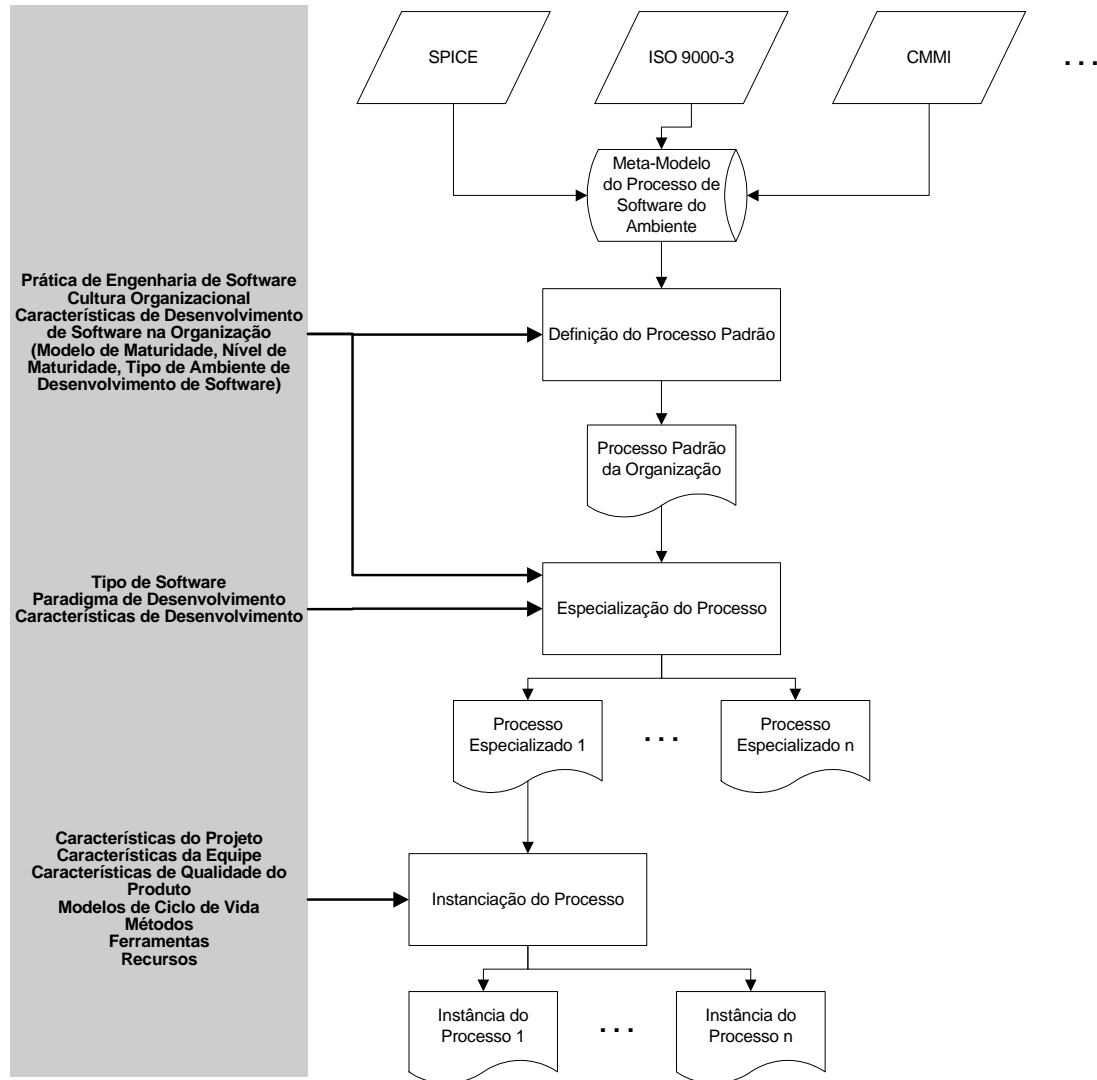


Figura 3: Estrutura de Adaptação do Processo em Níveis de Caracterização

3. Relato da Necessidade do Estudo de Caso

Embora a estrutura apresentada do ambiente possa estar caracterizada em um nível de completude capaz de atender as necessidades de todas as atividades do ciclo de vida de um processo de software, esta ainda não foi validada a partir de um estudo de caso real. Sendo apenas analisada sua importância e viabilidade a partir das características que compõem os ambientes citados na seção 1.

No entanto, planeja-se, como ação futura, uma experimentação das idéias propostas neste ambiente com base em um roteiro definido como eficaz pelo grupo de pesquisa, aliado ao uso de algumas

empresas regionais que tenham como foco o desenvolvimento de produtos de software.

Este roteiro consiste nos seguintes passos: análise das características organizacionais e dos tipos de projetos de software desenvolvidos; definição do processo de software com base em sugestões do ambiente e seguindo os três níveis da estrutura representada pela Figura 3; modelagem diagramática deste processo para uma melhor visualização dos seus componentes; instanciação do mesmo usando-se propriedades (tempo, pessoas, custo, etc.) de um projeto de software; captura das lições aprendidas a fim de propiciar a melhoria contínua e a reutilização de processos definidos; simulação do processo usando-se um ferramenta de suporte; execução do processo a partir de uma

máquina de inferência; avaliação desta execução para uma análise do que fora implementado.

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

4. Considerações Finais

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

No entanto, observou-se que a qualidade do produto de software está ligada ao processo de geração do software. Espera-se que ao longo dos anos as organizações de desenvolvimento de software ajustem seus processos de software para a produção de produtos de qualidade dentro de prazos confiáveis. Mais ainda, estas organizações serão pressionadas constantemente a otimizar os seus processos de desenvolvimento e manutenção, de forma a produzir produtos cada vez a custos menores e com qualidade crescente.

Assim, um ambiente capaz de prover a implementação progressiva de processos de software a partir da definição, simulação, execução e avaliação deste processo é de fundamental importância para que o cenário de aperfeiçoamento dos processos das organizações de desenvolvimento de software seja beneficiado com a automação das atividades.

Atualmente o ambiente encontra-se em fase de especificação de cada um dos módulos definidos na Figura 1, para que futuramente estes possam ser desenvolvidos e validados a partir de projetos reais, usando como estratégia a área de negócios de uma empresa regional que foca na definição, execução e avaliação de processos de software a partir de modelos/normas de qualidade para processo de software.

Referências

Bandinelli, S. et al. (1994) “SPADE: An Environment for Software Process Analysis,

Design and Enactment”, In: FINKELSTEIN, A. et al. (Ed.). Software Process Modelling and Technology. Taunton: Research Studies Press.

Balduino, R. (2002) “Implementação de um processo de desenvolvimento de software: uma abordagem passo-a-passo”, Rational Software White Paper.

Belkhatir, N., Estublier, J., Melo, W. (1994) “ADELE-TEMPO: an Environment to Support Process Modelling and Enaction”, In: FINKELSTEIN, A. et al. (Ed.). Software Process Modelling and Technology. Taunton: Research Studies Press.

Chrissis, M. B., Konrad, M. and Shrum, S., CMMI Guidelines for Process Integration and Product Improvement, Addison-Wesley, 2003.

Christie, A. (1995) “Software Process Automation: The Technology and its adoption”, Berlin: Springer Verlag.

Falbo, R. A. (1998) “Integração de Conhecimento em um Ambiente de Desenvolvimento de Software”, Orientadora: Ana Regina Cavalcanti da Rocha. Tese de Doutorado, COPPE/UFRJ.

Gimenes, I.M.S. (1994) “O Processo de Engenharia de Software: Ambientes e Formalismos”, Caxambu-MG: SBC, 1994. Trabalho apresentado na Jornada de Atualização em Informática.

ISO/IEC TR 9000-3 (1997) “The Application of ISO 9000 Series Standards to Software – Guidelines in Plain English”, International Organization for Standardization.

ISO/IEC TR 15504, Parts 1-9 (1998) “Information Technology – Software Process Assessment”, International Organization for Standardization.

Junior, J. B. (1997) “Qualidade de Software”, Notas de Aula do Curso de Qualidade de Software: Instituto de Software do Ceará.

Mcfeeley, B. (1996) “IDEALSM: A User’s Guide for Software Process Improvement”, Software Engineering Institute Handbook. Carnegie Mellon University. CMU/SEI-96-HB-001.

Nguyen, M. N.; Wang, A. I. (1996) “Total Software Process Model in EPOS”, disponível por [www](http://www.idt.unit.no/~epos/Papers) em: <http://www.idt.unit.no/~epos/Papers>.

Paulk, M. C., Curtis, B., Chrissis, M. B. and Weber, C. V. (1993) “Capability Maturity Model for Software”, Version 1.1. Technical Report CMU/SEI-93-TR-024. Software

Engineering Institute - Carnegie Mellon University.

Proforma (2000) “Business Process Improvement”, disponível por [www](http://www.proformacorp.com/downloads/whitepapers.asp) em: <http://www.proformacorp.com/downloads/whitepapers.asp>.

Reis, C. A. L. (2000) “Ambientes de Desenvolvimento de Software e seus Mecanismos de Execução de Processos de Software”, Orientador Daltro Nunes. Exame de Qualificação do Doutorado. PPGC-UFRGS.

Reis, C. A. L. (2003) “Uma Abordagem Flexível para Execução de Processos de Software Evolutivos”, Tese de Doutorado. PPGC-UFRGS.

Rocha, A. R. C., Maldonado, J. C. and Weber, K. C., Qualidade de software: teoria e prática, São Paulo: Prentice-Hall, 2001.

Werner, C. M. L., Mangan, M. A. S., Murta, L. G. P., Souza, R. P., Mattoso, M., Braga, R. M. M., Borges, M. R. S. (2003) “OdysseyShare: an Environment for Collaborative Component-Based Development”, In: Information Reuse and Integration (IRI), Las Vegas, Nevada, USA.