

## FERRAMENTA DE WORKFLOW DE DOCUMENTOS PARA O AMBIENTE COLABORATIVO ARCASE

**Marcello Thiry**  
thiry@univali.br

**Ana Frida da Cunha Silva**  
anafrida@univali.br

*Universidade do Vale do Itajaí – UNIVALI Campus São José  
LQPS – Laboratório de Qualidade e Produtividade de Software  
Rodovia SC 407, km 4  
88122-000 São José-SC Brasil  
Fone: (048) 281-1505 Fax: (048) 281-1506*

### Resumo

Apesar da importância inerente do setor de software, muitos projetos ainda são executados sem um processo estabelecido de desenvolvimento. A escolha adequada de um processo de desenvolvimento irá permitir o gerenciamento de riscos e um controle maior sobre a qualidade do produto. Sendo assim, a falta de um processo pode resultar em falhas na entrega, na estimativa de tempo e custo e na insatisfação do cliente. Neste sentido, um ambiente colaborativo centrado em casos de uso que suporta a fase de concepção de um processo iterativo vem sendo desenvolvido. O ambiente ARCASE é resultado de projetos anteriores e de trabalhos de conclusão do curso de Ciência da Computação da UNIVALI. Ele é baseado em um processo de software intermediário podendo ser utilizado por equipes de porte reduzido. Este trabalho corresponde à implementação do módulo de workflow de documentos do ARCASE, onde seu objetivo é oferecer uma ferramenta para gerenciar a tramitação dos artefatos produzidos pelo ambiente. Este módulo proporciona a colaboração entre os participantes, estendendo-se aos clientes que passam a ser co-responsáveis pelos artefatos em produção. Seu desenvolvimento segue o próprio processo definido para o ARCASE e é baseada em componentes estruturados em uma arquitetura de software em camadas (MVC).

**Palavras - Chave:** Processo de software, Casos de Uso, Workflow.

### 1. Introdução

Segundo pesquisas internacionais, o setor de software será responsável pelos maiores índices de crescimento na economia global nos próximos anos (ASSESPRO/RJ, 2004). Esta tendência pode ser observada também no Brasil, onde o setor de informática cresceu em média 20% entre 1995 e 2000, com uma comercialização bruta de R\$ 25,6 bilhões em 2000, sendo R\$ 5,9 em software. Este volume de software representa 1,3% do mercado mundial (SOFTEX/MIT, 2003). Essa situação favorável incentiva a criação constante de novos empreendimentos ligados a esta área, gerando uma maior demanda por critérios de qualidade e produtividade que permitam às empresas de software ganharem maior competitividade no mercado.

O desenvolvimento de software é um negócio que precisa ser produtivo para suportar a contínua competição. Entretanto, as empresas de sucesso estão modificando seu enfoque com relação ao conceito de produtividade. Segundo Armour e Miller (2000), os novos métodos para fazer negócios são primariamente baseados na informação e na tecnologia, sendo dirigidos pela competição global. Esta nova era, na qual métodos de produção são baseados mais no conhecimento do que no trabalho e competição, tem sido definida como a idade da informação.

Entretanto, apesar da informação e da tecnologia serem importantes, elas não são necessariamente suficientes para suportar uma modificação desta escala no mundo dos negócios. As empresas precisam buscar formas de utilizá-las para aumentar sua produtividade e vantagem competitiva. Peter Drucker (1994), já previa que a produtividade do conhecimento seria um fator determinante na posição competitiva de uma empresa.

O impacto e a rápida evolução ao longo dos últimos 40 anos das tecnologias relacionadas com os sistemas de informação têm colocado sucessivos desafios às empresas. A dependência e demanda crescentes da sociedade em relação à Informática e, em particular, a software, tem ressaltado uma série de problemas relacionados ao processo de desenvolvimento de software: alto custo, alta complexidade, dificuldade de manutenção, e uma disparidade entre as necessidades dos usuários e o produto desenvolvido. Neste período, vários modelos foram propostos e estabelecidos para diferentes equipes de desenvolvimento. Entretanto, muitos destes modelos não são adequados para pequenos grupos de desenvolvedores, dada a sua complexidade e alto custo de implantação.

Atualmente, a prática tem mostrado resultados que estimulam a revisão dos procedimentos tradicionais de engenharia de requisitos. O trabalho em grupo é outra importante ferramenta para aumentar o conhecimento sobre um determinado assunto. O processo de discussão faz com que novas idéias sejam consideradas e oferece diversas visões sobre o mesmo problema. Estas novas tendências de trabalho são interessantes na área de desenvolvimento de software, uma vez que a característica de resolver problemas de forma colaborativa (em grupo) é uma realidade (BECK, 2000; COCKBURN, 2000).

Neste sentido, na última década, um segmento crescente da comunidade de Engenharia de Software vem defendendo a existência de problemas fundamentais da aplicação sistemática e institucionalizada de processos de software convencionais (HIGHSMITH, 2002; BECK, 2000; FOWLER, 2001). Estes proponentes advogam processos simplificados, focados nas pessoas que compõem o processo, e principalmente no programador.

Seguindo esta tendência, este projeto é focado em um processo de software adaptado a partir do trabalho de Rosenberg e Scott (1999). O processo pode ser considerado como sendo intermediário, pois não possui a complexidade dos processos mais tradicionais e, ao mesmo tempo, possui um grau de formalidade superior aos processos considerados ágeis. O processo foi resultado de refinamentos resultantes da aplicação prática de projetos de grande porte, realizados nos últimos anos pela equipe do LQPS (Laboratório de Qualidade e Produtividade de Software) da UNIVALI Campus São José. Este processo tem alcançado resultados bastante positivos tanto para a equipe técnica como para os clientes envolvidos. Vale a pena ressaltar que o processo foi aplicado por equipes inexperientes na construção de artefatos formais. Isto também caracteriza o processo como sendo de fácil aprendizado. Além disso, o processo pode ser adaptado para equipes de pequeno e médio porte, sendo, portanto, aplicável a grande parte das empresas de software nacionais.

Atualmente, a fase de concepção do processo já está disponível por meio de um ambiente também desenvolvido no LQPS. O ambiente, denominado ARCase (KONOWALOW, 2003; DOMINGOS, 2004) suporta a elaboração dos principais artefatos que compreendem a fase de concepção, estando centrado em casos de uso e oferecendo alta rastreabilidade. Entretanto, o ambiente não oferecia nenhum recurso de tramitação para os artefatos construídos, durante um possível processo de avaliação e validação. Esta tramitação torna-se importante para permitir a efetiva participação dos clientes, garantindo maior transparência sobre a evolução dos artefatos.

Sendo assim, este trabalho apresenta uma solução integrada ao ambiente ARCase para suportar o controle de fluxo de artefatos, oferecendo recursos para validação e acompanhamento. A solução implementada adota a metáfora de uma caixa postal, onde os artefatos são recebidos, avaliados e enviados para o próximo destinatário (participante do projeto). Desta forma, é possível que qualquer usuário, inclusive os clientes, possa acompanhar a evolução de um artefato, contabilizando medidas como número de interações e tempo de avaliação em cada avaliador.

## **2. Processo de Desenvolvimento de Software**

Atualmente, no contexto de uma indústria, a qualidade de um produto de qualquer natureza não pode ser resultado exclusivo da habilidade de um artesão. Isto limitaria as linhas de produção e dificultaria a verificação do produto em relação aos seus requisitos iniciais. Desta forma, torna-se necessário documentar, organizar e detalhar as atividades que devem ser executadas para a construção do produto para garantir que cada produto tenha sido resultado de um mesmo processo. Este processo também irá permitir a definição de pontos de controle, que irão facilitar o acompanhamento da execução das atividades e o gerenciamento dos riscos envolvidos.

Na produção de software, a situação é a mesma. Entretanto, as características inerentes do software o tornam um produto de difícil definição e medição. Desta forma, os processos para a construção de sistemas não podem utilizar diretamente aqueles mesmos processos já consolidados na indústria em geral. Aliás, esta foi à idéia inicial da Engenharia de Software (GHEZZI *et al*, 1991; CANTOR, 1998), mas logo se verificou a necessidade de adaptações devido às dificuldades impostas pela forma de desenvolver sistemas.

A definição de processo de software é bastante discutida, havendo na verdade, uma certa sobreposição em relação aos termos Processo, Modelo, Método e Metodologia. Esta discussão não faz parte do contexto deste trabalho e pode ser vista em (BONA, 2002). Entretanto, é necessário estabelecer o conceito de processo na visão dos autores deste projeto.

Um processo de software pode ser entendido como uma seqüência de fluxos de trabalho, onde para cada fluxo, é definida uma série de atividades. Estas atividades serão executadas por um ou mais participantes e irão gerar um ou mais artefatos. Um artefato pode ser entendido como sendo qualquer resultado tangível durante a execução da atividade. Cada participante irá assumir diferentes responsabilidades que são associadas de acordo com atividade. Os requisitos do sistema irão formar a entrada do processo que, após a execução da seqüência dos fluxos de trabalho, irá ter como saída, o sistema. Outros conceitos podem ser encontrados em Sommerville (1995), Jacobson *et al* (1999) e Cantor (1998).

O processo adotado e adaptado para este trabalho utiliza um ciclo de vida iterativo e incremental (JACOBSON, BOOCH e RUMBAUGH, 1999), no qual os fluxos de trabalho são distribuídos em quatro fases distintas: concepção, elaboração, construção e transição. O foco desta breve revisão está na fase de concepção.

### **2.1. Fase de Concepção**

A fase de concepção tem como propósito o entendimento inicial e a concordância da definição do produto, isto é, o que deve ser entregue. A principal atividade é a revisão do escopo do projeto, através de uma modelagem de negócio e revisão dos requisitos do sistema. O principal artefato

desta fase é um documento com a visão geral do sistema em desenvolvimento. Segundo Jacobson, Booch e Rumbaugh (1999), esta fase procura:

- Identificar e reduzir aqueles riscos que são críticos para a viabilidade do sistema.
- Mapear o caminho a partir de um conjunto de requisitos através da modelagem de casos de uso em uma arquitetura de software.
- Realizar uma estimativa inicial de custos, esforço, escalonamento, qualidade, considerando limites bem definidos.
- Iniciar a construção do modelo de negócio, considerando os limites estabelecidos.

No contexto deste trabalho, a fase de concepção possui ainda mais importância, pois ela avança no detalhamento dos requisitos, incluindo também a modelagem detalhada das regras de negócio. O elemento principal do documento com a visão geral do sistema é o modelo de caso de uso. Entretanto, os casos de uso capturam apenas a funcionalidade de um sistema. Desta forma, torna-se necessários artefatos suplementares que irão suportar as características da fase de concepção. É dentro deste contexto que o ambiente ARCase atua, oferecendo um padrão de preenchimento para os diversos artefatos e garante a rastreabilidade dos mesmos. Por estarem fora do escopo deste artigo, as demais fases não serão abordadas.

### **3. Workflow de Documentos**

Os casos de uso são os principais documentos da fase de concepção, uma vez que eles permitem capturar a funcionalidade do sistema. Uma característica importante de um caso de uso é a sua estrutura. Usualmente, eles são organizados em textos descritivos que definem vários fluxos dentro de um contexto. Como todo texto escrito em linguagem natural, os casos de uso podem apresentar ambigüidades e falhas de interpretação daquilo que foi levantado. É neste momento que se faz necessário um processo de avaliação e validação por parte dos clientes, na qual são eles que devem aprová-lo na sua forma final.

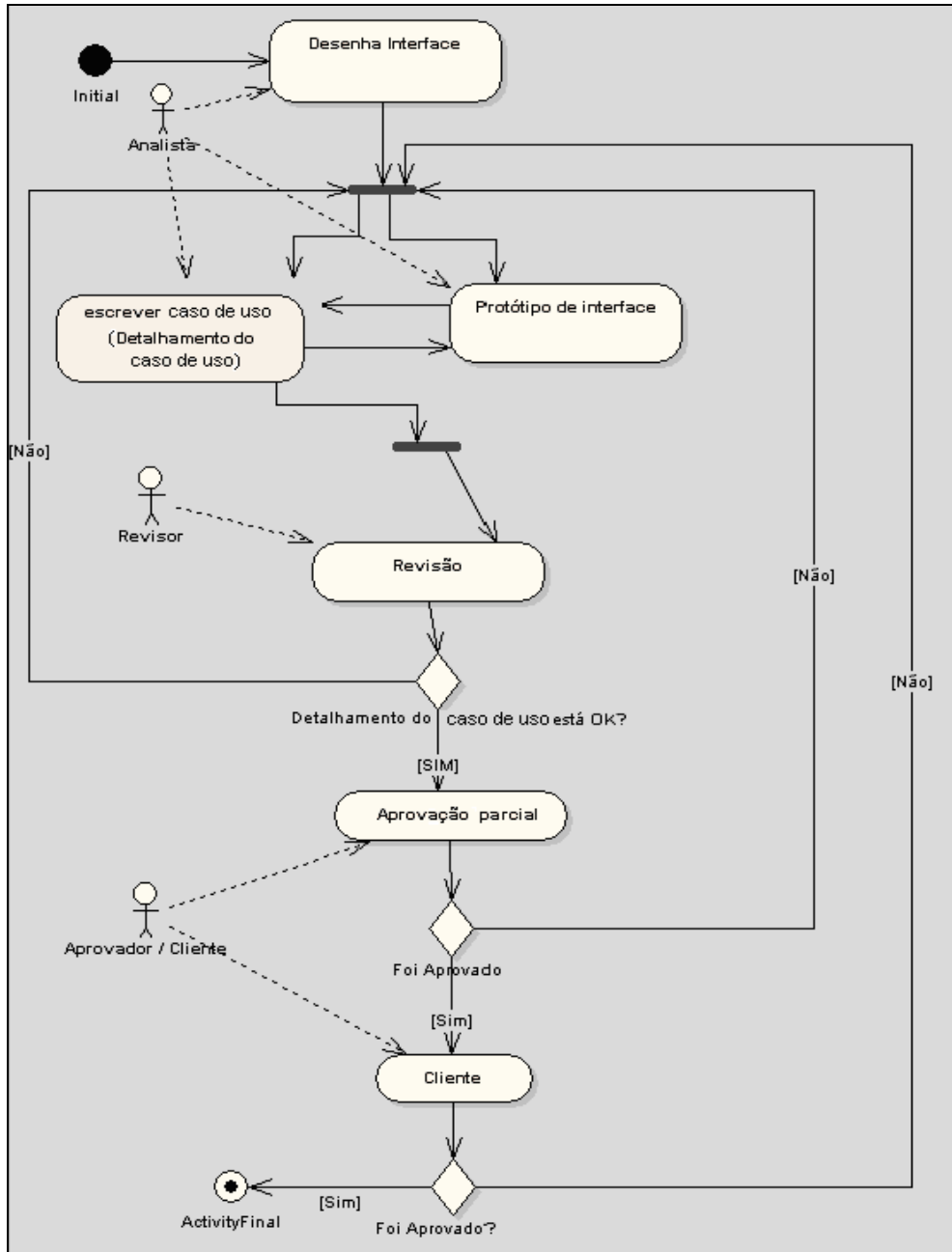
A partir da demanda em obter um maior controle sobre os fluxos produzidos no processo de tramitação dos casos de uso, desenvolveu-se uma ferramenta para gerenciar essa tramitação com os artefatos produzidos pelo ambiente ARCase. O workflow define uma seqüência ordenada de atividades que devem ser seguidas para a realização de uma determinada tarefa no processo. Processo é um conjunto de atividades que tem por objetivo atingir um resultado. Para a execução destas atividades, associam-se papéis que representam a responsabilidade de cada participante envolvido. Cada workflow pode requerer a interação de diversos participantes, cada um deles com uma responsabilidade definida.

Para Sutt (1996 *apud* ARAUJO) e Fisc (1997 *apud* ARAUJO), os sistemas de workflow tiveram sua origem na década de 70 a partir de pesquisas para organizar os escritórios que tinham como objetivo, armazenar, compartilhar e tramitar documentos nas organizações para reduzir a quantidade de documentos em papel. Atualmente, o workflow é estratégia para se obter melhorias e assim proporcionar um controle maior dos fluxos existentes no processo.

#### **3.1 Modelo de Workflow Adotado**

A figura 1 apresenta um exemplo de fluxo no módulo implementado. Para a representação, adotou-se o diagrama de atividades da UML. Neste exemplo, o fluxo define a validação de um caso de uso.

O fluxo de atividades inicia pela prototipação da interface, ou seja, as telas relacionadas ao caso de uso. Estas telas são elaboradas pelo Analista, que também possui todo o direito em descrever os casos de uso, na qual essas duas tarefas podem ser feitas paralelamente.



**Figura 1.** Fluxo para aprovação de um caso de uso.

Ao concluir a descrição do caso de uso, este é submetido à revisão de um revisor. O revisor verifica toda a descrição do caso de uso. Se o revisor não concordar ou ficar com dúvida sobre a descrição do caso de uso, este é retornado ao analista para que sejam feitas as alterações solicitadas. Caso contrário, o caso de uso é enviado ao cliente que iniciará a validação. Se o cliente não concordar com alguma parte do caso de uso, ele é retornado ao analista para que sejam feitas as modificações necessárias. Se o caso de uso estiver de acordo com os critérios de avaliação do cliente, então, este fluxo de validação é finalizado e o caso de uso passa a receber o status de aprovado.

## 4. O Ambiente ARCase

O ambiente ARCase trata-se de uma ferramenta colaborativa para a fase de concepção de software. Dentre as suas funcionalidades esta o módulo de workflow de artefatos, cujo principal foco do trabalho é relatado neste artigo. Na figura 2 é apresentada a tela de menu principal do sistema. A partir dela, é possível cadastrar informações sobre os projetos, participantes, além do acesso aos serviços da ferramenta.

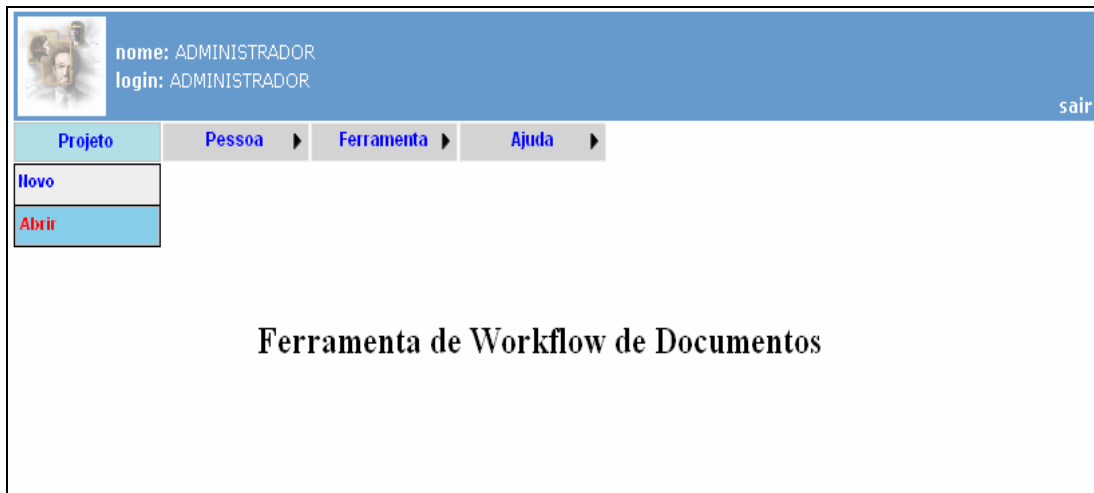


Figura 2. Tela de Menu do sistema.

Para dar início ao processo de workflow, é feita a associação de um determinado avaliador com um determinado artefato no processo de avaliação conforme apresentado na figura 3. Esta etapa independe do caso de uso estar apto ou não para avaliação.

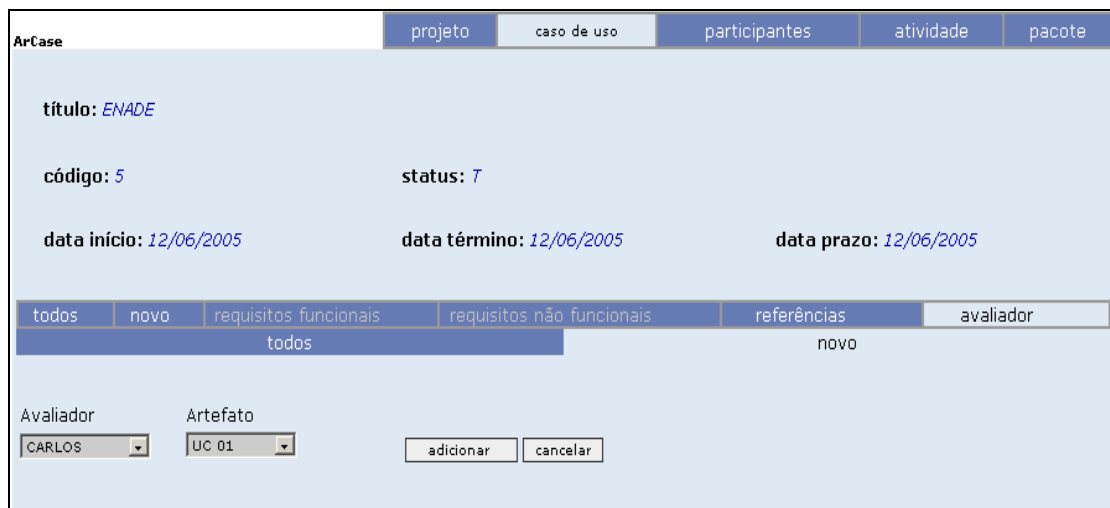


Figura 3. Tela de associação de um determinado avaliador com um determinado artefato.

O responsável pela edição do artefato utiliza a tela apresentada conforme figura 4 que neste caso, contém toda a descrição do artefato, além de estar apto para ser avaliado. Após estar apto para ser avaliado, ele é enviado ao avaliador para dar início ao processo de avaliação e validação do artefato.

Figura 4. Tela de edição de caso de uso.

Após o envio dos artefatos aos seus respectivos avaliadores, o avaliador visualiza a lista contendo o código e a descrição de todos os artefatos a serem avaliados e seleciona qual artefato deseja avaliar conforme apresentado na figura 5.

codigo	descrição	
UC 01	Indui Avaliador	abrir
UC 02	Exclui Avaliador	abrir

Figura 5. Tela Listar Todos casos de uso a serem avaliados pelo usuário.

Selecionado, o artefato é visualizado pelo avaliador conforme apresentado na figura 6 e neste sentido, o avaliador observa toda a descrição do artefato e decide se aprova ou aponta alguma observação no artefato avaliado.

ArCase

projeto caso de uso participantes atividade pacote

título: ENADE

código: 5 status: 7

data início: 12/06/2005 data término: 12/06/2005 data prazo: 12/06/2005

todos visualizar requisitos funcionais requisitos não funcionais referências

autor: ADMINISTRADOR pacote: 1 apto para avaliação: sim

descrição do caso de uso:

<CODIGO>UC 01  
 <DESCRICAO>Inclui Avaliador  
 <ATORATIVO>Administrador  
 <BASE>  
 1. O sistema apresenta a tela Avaliador (TEL. 01)  
 2. O administrador seleciona um avaliador e um artefato  
 3. O administrador seleciona a opção adicionar  
 4. O sistema associa o avaliador com o artefato

<ALTERNATIVO> Se o administrador selecionar a opção cancelar  
 3.1 O sistema apresenta a Tela Lista Avaliador (TEL. 02)

aprova tem restrição

**Figura 6.** Tela de Visualização do Caso de uso.

Se o avaliador não concordar com a descrição do artefato, o avaliador descreve as suas observações conforme apresentado na figura 7 e reenvia essas observações ao responsável pela edição do artefato. Caso o avaliador concorde com a descrição do artefato, este artefato é aprovado e o processo de tramitação finalizado.

Observações

E se no passo 3, o administrador desejar selecionar a opção cancelar. Este caso de uso não trata este fluxo alternativo.

Gravar comentários cancelar

**Figura 7.** Tela de Observações do usuário.

Os trâmites são gerados para cada artefato cadastrado no ambiente ARCase, obtendo assim um controle maior desses artefatos gerados pela própria ferramenta.



## 5. Conclusão

Este trabalho apresentou o desenvolvimento de um módulo para controlar workflows de documentos dentro do ambiente colaborativo ARCase. Este ambiente vem sendo desenvolvido por projetos de iniciação científica e trabalhos de conclusão de curso sob a orientação do LQPS/UNIVALI. Sua versão atual foi desenvolvida totalmente em JAVA para a web, utilizando o banco de dados MySQL.

Atualmente, o ARCase possui recursos para suportar a fase de concepção de software sendo centrado em casos de uso. Ele oferece alta rastreabilidade entre os artefatos, definindo um template para a documentação dos casos de uso. O template é organizado por TAGs que identificam partes e conexões de um caso de uso (figura 4).

Para a construção do módulo de workflow, foram necessárias adaptações do ambiente. O mecanismo de associação entre os participantes e os respectivos papéis teve que ser modificado para permitir a definição de clientes como avaliadores. Foi também desenvolvido um serviço para armazenar a seqüência prévia de trâmites pela qual um tipo de artefato deverá passar. O módulo de workflow implementado utiliza a metáfora de uma caixa postal, permitindo que qualquer artefato possa tramitar entre os participantes. Durante a tramitação, os avaliadores podem introduzir comentários que estarão disponíveis para os analistas. Desta forma, é possível implementar um serviço de validação em tempo real com os clientes de um projeto de software, antecipando quaisquer tipos de interpretação errada que possam ter ocorrido durante o levantamento dos requisitos.

Para permitir sua validação, o ambiente ARCase está iniciando seu processo de validação com turmas de Engenharia de Software que estarão simulando equipes de levantamento e documentação de requisitos. Desta forma, as modificações feitas no cadastramento de casos de uso e dos artefatos relacionados (telas do sistema, regras de negócio, requisitos funcionais e não funcionais) poderão ser verificadas, assim como a usabilidade da ferramenta. Para avaliar a funcionalidade da estrutura de workflow proposta, será utilizada uma revisão pelos pares, onde cada equipe assumirá o papel de cliente para avaliar os artefatos escritos por outra equipe.

O ambiente ARCase é uma contribuição para a área de projeto de desenvolvimento de software, não apenas pelo seu aspecto de colaboração, mas por ser uma ferramenta prática que permite armazenar de forma eficiente todo o processo de tramitação de artefatos gerados por ela, cujos artefatos são os próprios casos de uso. Entretanto, o trabalho não está concluído. Atualmente, o LQPS está desenvolvendo a plataforma PLACES, a qual é uma plataforma colaborativa para suportar atividades de Engenharia de Software. Além das características colaborativas que estão sendo migradas do ambiente ARCase, a plataforma oferecerá uma ferramenta de gerência de projetos e o atual ambiente ARCase passará a ser a ferramenta que suportará os processos de definição e gerência de requisitos. Desta forma, a solução apresentada aqui para gerenciar um workflow de documentos poderá ser generalizada para atender não apenas a fase de concepção, mas todas as fases de um processo de desenvolvimento de software.

## 6. Referências

- ARAUJO, Renata M. de e BORGES, Marcos R. da Silva. **Sistemas de Workflow**. Congresso da SBC. Departamento de Ciências da Computação/ Instituto de Matemática. UFRJ, 2001. Disponível em: <<http://chord.nce.ufrj.br/cursos/teesi/textos/apostilaJai2001div.pdf>>. Acesso em: 18 abr. 2005.
- ARMOUR, F. e MILLER, G. **Advanced Use Case Modeling: Software Systems**. Addison-Wesley, Object Technology Series, 2000.
- ASSESPRO/RJ. Associação das Empresas de Processamento de Dados do Estado do Rio de Janeiro. **Propostas Relativas ao Marco Tributário do Setor de Software**. <http://www.assespro-rj.org.br/instituc/politicaTrib.html>, 2004.
- BECK, K. **Extreme Programming Explained: Embrace Change**. Addison-Wesley, 2000.
- BONA, C. **Avaliação de Processos de Software: Um Estudo de Caso em XP E Iconix**. Dissertação de Mestrado. PPGEP/UFSC, 2002.
- CANTOR, M. **Object-Oriented Project Management with UML**. Wiley & Sons, 1998.
- COCKBURN, A. **Crystal Methodologies**, 2001. Disponível em: <<http://crystalmethodologies.org/index.html>>. Acesso em: 18 abr. 2002.
- DOMINGOS, M. **Um editor de casos de uso para suportar a rastreabilidade de artefatos em um processo de software**. Trabalho de Conclusão de Curso. Ciência da Computação. UNIVALI, Campus São José, 2004.
- DRUCKER, P. **Pos-Capitalist Society**. HarperBusiness, maio, 1994.
- FOWLER, Martin. The New Methodology, nov, 2001. Disponível em: <<http://www.martinfowler.com/articles/newMethodology.html>>. Acesso em: 18 abr. 2002.
- GHEZZI, C. e JAZAYERI, M. e MANDRIOLI, D. **Fundamentals of software engineering**. Prentice-Hall, 1991.
- GIL, A. C. **Como elaborar projetos de pesquisa**. 3.ed. São Paulo: Atlas, 1996.
- HIGHSMITH, Jim. Does Agility Work? Software Development. Canadá, v.10, n.6, p.30-36, jun. 2002.
- JACOBSON, I. e BOOCH, G. e RUMBAUGH, J. **The Unified Software Development Process**. Addison-Wesley, 1999.
- KONOWALOW, A. **Um ambiente colaborativo de suporte à fase de concepção de software**. Relatório Final de Projeto de Iniciação Científica. PIBIC/CNPq. UNIVALI, Campus São José, 2003.
- ROSENBERG, D. e SCOTT, K. **Use Case Driven Object Modeling with UML: A Practical Approach**. Addison-Wesley, 1999.
- SOFTTEX/MIT. **Pesquisa realizada pela Sociedade para Promoção da Excelência do Software Brasileiro e pelo Massachusetts Institute of Technology**. <http://www.softex.br/cgi/cgilua.exe/sys/start.htm?inford=2955&sid=42>, 2003.
- SOMMERVILLE, Ian. **Software Engineering**. Lancaster University, Lancaster: Ed. Addison-Wesley, 1995.