

# **ProReuso: Um Repositório de Componentes para Web Dirigido por um Processo de Reuso**

Caroline Batista Spencer Holanda  
Clarissa Angélica de Almeida de Souza  
Walcélio L. Melo  
walcelio.melo@oracle.com

## **Resumo**

Alcançar o máximo de qualidade, produtividade e eficiência no desenvolvimento de sistemas é um dos objetivos fundamentais da Engenharia de Software. O reuso de software de forma sistematizada, ou seja, direcionada para as metas do negócio, através de uma estratégia técnica e gerencial, torna possível a realização destes objetivos. Nesse contexto, um repositório, atuando como um mecanismo de armazenamento e recuperação, contribui para se alcançar as metas propostas. Objetivando o incentivo da prática do reuso de software nos processos de desenvolvimento e visando demonstrar a sua viabilidade, é apresentado neste projeto um modelo de repositório para web, denominado ProReuso, onde é possível o armazenamento, a análise, a classificação e a recuperação contextualizada de componentes e documentos relativos ao processo de construção de softwares.

## **Palavras-chaves**

Engenharia de Software; Qualidade de Software; Reuso; Repositório de componentes; Ambientes de Desenvolvimento Dirigido por Processos; Internet; Workflow.

## **ProReuso: A WEB Component Repository Driven by a Software Reuse Process**

Caroline Batista Spencer Holanda  
Clarissa Angélica de Almeida de Souza  
Walcélio L. Melo  
walcelio.melo@oracle.com

## **Abstract**

One of Software Engineering's fundamental objectives is to reach the best in quality, productivity and efficiency in systems development. Software reuse in a systematized way, or else, directed to business goals through a managing and technical strategy, allows the achievement of this objective. In this context, a repository, acting as a store and search mechanism, contributes for the achievement of the proposed goals. Aiming at the encouragement of software reuse practice in development processes and at the demonstration of its viability, a web repository, denominated ProReuse(where it is possible to store, analyze, classify and contextually recuperate components and documents related to software building process), is described.

## **Key-Words**

Software Engineering; Software Quality; Software Reuse; Component Repository; Process-Driven Software Engineering Environments; Internet; Workflow.

# **1 Introdução**

## **1.1 Motivação**

O reuso propõe o desenvolvimento de sistemas a partir de produtos de software preexistentes, de modo que semelhanças entre requisitos e arquiteturas de diversos sistemas possam ser melhor exploradas [FAVARO et al., 1998] e [BASILI et al., 1996]. A identificação dessas semelhanças pode resultar, entre outras coisas, significativas reduções de custo e tempo, isto é, aumento de produtividade para as empresas desenvolvedoras. Além disso, a cada uso do produto, sua eficiência e validade são verificadas, garantindo, assim, maior qualidade e redução de erros nos produtos finais.

Um repositório de componentes, por sua vez, representa uma ferramenta em potencial para a prática do reuso dentro das organizações. Quando os componentes nele armazenados são bem organizados, divulgados e gerenciados, esse potencial torna-se realidade. Através de um repositório sustentando as políticas de reuso dentro da empresa, tendo em vista os benefícios dessa prática, conseqüentemente haverá aumento nos níveis de produtividade e qualidade no processo de desenvolvimento da organização que o utiliza. Ou seja, reuso de software é uma estratégia de negócio que deve ser sustentada, do ponto de visto tecnológico, por um repositório de componentes corporativos.

Assim, diante dessas considerações e levando-se em conta o mercado cada vez mais competitivo no qual as empresas de software estão inseridas, a implementação de um repositório de componentes de software que viabilize o processo de reuso torna-se um fator diferencial que permite a elas entrar para o grupo daquelas que têm acesso aos benefícios trazidos pelo reuso.

## **1.2 Objetivo**

O objetivo deste artigo é descrever a experiência de um repositório de componentes dirigido por um processo de reuso.

Define-se, nesse trabalho, que o reuso é uma prática sistemática de desenvolver novas aplicações a partir de ativos de software pré-construídos que apresentam similaridades de requisitos e/ou arquiteturas com o que está sendo desenvolvido [KRUEGER,1992]. Por se tratar de uma prática sistemática requer disciplina e envolvimento de todos os membros da empresa. Além disso, requer um processo de conscientização interno, para que todos entendam o potencial de aumento de produtividade, qualidade dos produtos, economia em relação aos custos e tempo de desenvolvimento que essa prática representa.

Na definição de reuso, nós utilizamos a palavra ativo de software. Ativo de software é definido como aquilo que se pode reusar, que consiste em qualquer produto de software reusável, de qualquer fase do desenvolvimento [EZRAN et al., 1999]. Essa escolha ocorreu pelo fato do termo trazer com ele a idéia de valor e investimento para a empresa. Como essas idéias correspondem ao conceito de software reusável que se deseja deixar claro nesse projeto, o termo ativo está bem empregado.

Em relação ainda ao termo ativo, é importante esclarecer que na fase de implementação do repositório esse termo foi substituído por componente de software. Essa substituição não representa o esquecimento ou até contradição do que foi dito no parágrafo anterior. Ela ocorreu pelo fato da empresa que motivou a construção desse repositório, Oracle do Brasil, ter a intenção, inicial, de construir um repositório focado em componentes executáveis e documentos relacionados a eles, tais como, manual de usuário, diagramas UML [BOOCH et al., 1999], apresentações técnicas, etc., que atendesse principalmente a sua área de

desenvolvimento. Pode-se citar outro fator menos relevante, que consiste no fato do termo componentes ser mais familiar para as pessoas, em geral, quando se refere a reuso de software. Assim, apesar da substituição, ainda pode-se pensar em ativos de software sendo armazenados no repositório.

O ProReuso é, portanto, é um repositório de componentes de software reusáveis, que consiste numa aplicação para web, disponibilizado para funcionários da Oracle do Brasil, através da Intranet da empresa.

Com o ProReuso, o funcionário pode tanto alimentar o repositório com novos componentes e/ou documentos relacionados a eles como localizar e recuperar outros já disponibilizados. Para permitir a alimentação, o ProReuso disponibiliza as funcionalidades de submissão de componentes e de documentos, que disponibiliza a entrada de características dos mesmos para que possam ser armazenados dentro de uma organização definida.

A fim de implementar um processo de gerenciamento daquilo que está sendo disponibilizado no repositório, é implementado no ProReuso um processo de aprovação de componentes<sup>1</sup>, através de um *workflow*. Esse processo representa um crivo de gerentes de projeto e especialistas de qualidade, pelo qual todo componente ou documento submetido ao repositório deve passar. Tem-se, portanto, uma garantia de que aquilo que está disponibilizado já foi verificado e aprovado por especialistas.

Para a eficiência na pesquisa de ativos disponibilizada no ProReuso, utilizou-se os recursos de estruturação textual oferecida pela tecnologia XML(*eXtensible Markup Language*) [GRAHAM, 1999]. Cada ativo possui um documento XML associado, que é alimentado durante a submissão, que armazena as informações do mesmo de forma estruturada, correspondendo às características do ativo. A indexação desses documentos é feita através de uma ferramenta (*Oracle Intermedia*) disponibilizada pelo banco de dados, utilizando-se a estruturação feita segundo as facilidades fornecidas pelo XML.

Assim, considera-se essas duas últimas funcionalidades diferenciais importantes disponibilizados no ProReuso. Dessa forma, acredita-se que o ProReuso representa um exemplo de repositório que serve, de fato, como ferramenta auxiliar para a prática de reuso dentro de uma organização.

## **2 Repositório: Reutilização de Componentes**

### **2.1 Introdução**

Com a crescente importância atribuída ao reuso de software e informações, muito se tem discutido a respeito da organização e centralização desses valores. O repositório surge como uma solução para se resolver um paradoxo observado nas organizações: sabe-se que o compartilhamento de experiências e informações pode proporcionar grandes vantagens competitivas, entretanto muitas empresas ficam impossibilitadas de aproveitar tais benefícios porque não conseguem lidar com a variedade e a distribuição ineficiente das informações desejadas e, muitas vezes, sequer têm conhecimento das informações disponíveis [HENNINGER 1997].

O termo repositório pode ser utilizado para diversos propósitos. Entre eles destacam-se:

1. Armazenamento de modelos – um processo de desenvolvimento envolve a geração de vários artefatos que normalmente são armazenados em locais diferentes. Um repositório

---

<sup>1</sup> Nesse processo, o termo componente é usado também para se referir a documentos, com intuito de ter uma coerência com o nome da aplicação.

possibilita a centralização do armazenamento desses artefatos e o controle de cada uma das suas versões.

2. Integração de ferramentas – muitas vezes os modelos gerados durante um processo de desenvolvimento originam-se de ferramentas distintas. Assim, para a troca de informações entre elas é necessário que haja uma “linguagem” comum através da qual as mesmas possam se comunicar. O repositório pode servir para um armazenamento temporário e para a intermediação entre as ferramentas, ou seja, pode transformar o modelo de entrada de uma delas num modelo compatível para a outra.
3. Manutenção de sistemas legados – um repositório pode ser útil também para o armazenamento de dados relacionados a sistemas antigos. Esses dados podem posteriormente ser analisados para se determinar os possíveis impactos gerados por modificações.

Analisadas tais funcionalidades, observa-se que o repositório é utilizado para o armazenamento de metadados. Essa restrição, conforme destaca [BABCOCK, 1998] é o que os diferencia dos bancos de dados.

Para o propósito deste trabalho, um repositório será considerado como o local onde os ativos de software estarão armazenados e classificados de modo a facilitar a correta utilização e recuperação dos mesmos.

## 2.2 Características de um Repositório

As principais funcionalidades de um repositório são as pesquisas e a recuperação dos ativos de software. No entanto, a presença de algumas outras funcionalidades pode facilitar e, com isso, incentivar a sua utilização. Conforme proposto por [EZTRAN et al., 1999], serão apresentadas, a seguir, algumas das principais funções que um repositório pode disponibilizar a seus usuários. Vale ressaltar que essas funções não são obrigatórias e variam dependendo do contexto no qual ele está inserido e das necessidades da organização ao utilizar um repositório.

**Identificação e descrição:** para se descrever um ativo é possível que a ele seja atribuído um conjunto de características tais como, nome, domínio, palavras-chave, dentre outras que os identificam e os diferenciam dos demais ativos que compõem esse mesmo repositório. Para cada um dos ativos armazenados deve ser possível identificá-lo homogeneamente dentro de um repositório, ou seja, os ativos de um mesmo tipo devem apresentar o mesmo conjunto de características. Isso não significa, contudo, que ativos diferentes devam ter o mesmos valores para esse conjunto de características.

**Inserção:** um repositório deve permitir que usuários autorizados insiram novos ativos ou, ainda, novas versões dos mesmos. A inserção significa adicionar ao repositório o corpo e a descrição do ativo, conforme apresentado no capítulo dois.

**Exploração do catálogo:** aos usuários de um repositório deve ser permitido que explorem o catálogo de ativos para que possam conhecer e analisar as características dos ativos disponíveis.

**Pesquisa textual:** um repositório deve permitir que seus usuários façam pesquisas mais específicas na descrição dos ativos. Como resultado da pesquisa serão obtidos um ou mais ativos que satisfaçam as condições desejadas. Observados os resultados, pode-se decidir por um maior detalhamento ou generalização dos critérios anteriores.

**Recuperação:** após a identificação do ativo desejado, um repositório deve permitir que seus usuários recuperem esse ativo para que possam posteriormente utilizá-lo num processo de reuso.

**Organização e pesquisa:** como visto anteriormente, a funcionalidade de exploração de catálogos não é suficiente à medida que a quantidade de ativos disponíveis aumenta. Além disso, a pesquisa textual, muitas vezes, demanda muito tempo. Devido a esses fatores, alguns repositórios podem adotar novas formas de organizar as características dos ativos de modo a permitir que a pesquisa no repositório seja baseada em outros critérios.

**Histórico:** é importante para o gerenciamento de um repositório que ele possa armazenar informações de uso, modificações, criação e exclusão de cada um dos ativos disponíveis. Essas informações devem montar uma base histórica que facilitará a análise e a reutilização dos mesmos.

**Mensuração:** um repositório pode coletar estatísticas que facilitem o seu gerenciamento. Algumas das principais estatísticas que podem ser adotadas são: frequência de acesso ao repositório, quantidade de ativos disponíveis, taxa de recuperação, porcentagem de pesquisas bem-sucedidas, frequência com que um determinado ativo é analisado ou modificado, dentre outras.

**Controle de acesso:** um repositório pode adotar uma política de segurança para que determinadas funcionalidades só estejam acessíveis a pessoas autorizadas. Por exemplo, pode-se definir para uma empresa específica uma política de segurança onde a pesquisa textual esteja disponível para todos os funcionários, mas a recuperação de ativos esteja disponível apenas aos funcionários da área de desenvolvimento.

**Gerenciamento de versões:** um repositório pode conter várias versões de um mesmo ativo e, sendo assim, é recomendável que haja algum mecanismo para controlar essas versões e estabelecer o relacionamento entre elas.

**Controle de modificações:** é recomendável que sejam providas algumas funcionalidades para se fazer o gerenciamento de modificações dos ativos num repositório. Essas funcionalidades incluem procedimentos para solicitação de alterações, discussões e permissão para as mesmas. É de extrema importância para a manutenção da consistência de um repositório que quaisquer intenções de modificar os ativos sejam comunicadas àquele responsável pela sua administração para que o mesmo, juntamente com outros responsáveis, possam analisar as alterações solicitadas e manter a coerência entre os diversos ativos de software.

**Notificação de mudanças:** é possível que a qualquer momento um ativo seja modificado ou, até mesmo, que o próprio repositório sofra algumas alterações em suas principais funcionalidades. Sendo assim, um repositório pode prover funções que notifiquem os seus usuários das modificações recentemente ocorridas tais como, inserção ou exclusão de ativos, alterações em documentos, disponibilização de novas funcionalidades, novas políticas de segurança, entre outras.

Apresentadas as principais funcionalidades de um repositório faz-se necessário analisar alguns dos aspectos não-funcionais que o caracterizam. Esses aspectos podem também, se adequadamente utilizados, possibilitar maiores incentivos para a utilização do repositório.

**Quantidade:** o repositório pode ser único ou pode existir um conjunto deles [EZLAN et al., 1999]. O único é indicado para organizações pequenas onde não há uma grande divisão departamental e, assim, os tipos de ativos não são completamente distintos e são compreendidos por todos os seus usuários. Para organizações maiores ou geograficamente distribuídas sugere-se o uso de vários repositórios, cada um relacionado a uma área de atuação

da empresa. Nesse caso é importante que haja um gerenciamento centralizado e eficiente para evitar inconsistência nas duplicações existentes entre alguns repositórios. A utilização de vários deles é defendida por aqueles que consideram que informações desnecessárias para um determinado grupo de usuários, mas de grande valia para um outro, prejudicam e desestimulam a prática do reuso nas organizações.

**Acesso pela rede:** como a maioria das empresas estão migrando seus modelos computacionais para modelos centralizados, um requisito típico é que o repositório possa ser acessado de qualquer ponto numa rede [EZTRAN et al., 1999]. Isso é particularmente importante quando seus usuários estão distribuídos geograficamente. O que normalmente se tem observado na implementação dos repositórios é a utilização de modelos cliente-servidor em 3 (três) ou mais camadas. No modelo de 3 camadas uma é a responsável pelo armazenamento dos ativos, a outra pelo processamento das principais funcionalidades do sistema tais como, pesquisa e recuperação e a terceira corresponde à aplicação cliente, por onde o repositório poderá ser acessado. A quantidade de camadas é o modo de implementação das mesmas poderão variar entre as diversas organizações.

### **2.3 Gerenciamento**

O gerenciamento de ativos representa descobrir, identificar e gerenciar cada ativo como uma oportunidade de reuso [UREP, 2000]. Conforme apresentado anteriormente muitas empresas possuem informações espalhadas por toda a organização. Cabe ao gerente de um repositório identificar aquelas que agregam valores à organização e incluí-las nesse repositório para que possam ser por todos compartilhadas.

Como o repositório representa uma coleção de ativos é necessário que haja um processo de gerenciamento para que se possa melhor controlar a quantidade e qualidade de ativos disponíveis, as suas localizações e as principais características desses elementos. Além disso, no caso de repositórios não-centralizados é importante que haja um rígido controle para se evitar inconsistências entre os diversos repositórios existentes.

O controle do catálogos de ativos é uma atividade imprescindível no gerenciamento de um repositório. Nele poderão ser encontradas referências para cada um dos ativos que o compõe e que serão fundamentais para a prática do reuso sistematizado. Sua ausência pode ser a maior contribuição para o caos [EZTRAN et al., 1999].

Como se pôde observar o real valor de um repositório está em conseguir localizar aquilo que se está procurando ou o que se adequa às necessidades do usuário. Assim, o gerenciamento deve contribuir para manter informações coerentes e adequadas a um determinado grupo de usuários. Isso inclui, como apresentado na seção anterior, controle de versões, controle de modificações e classificação adequada de cada um dos ativos disponíveis num repositório.

Outro ponto importante a ser observado no gerenciamento de um repositório é o controle de qualidade que deve ser imposto aos ativos que o constituirão. Finalmente, não basta que os ativos estejam disponíveis e possam ser localizados, mas é necessário, também, que eles possam ser compreendidos para, com isso, poderem ser reutilizados. Neste ponto destaca-se a importância da definição de padrões para os ativos e o papel dos gerentes na validação desses padrões.

### **2.4 Mercado de Repositórios**

Com a crescente motivação para o reuso de ativos de software nos processos de desenvolvimento, diversas empresas estão investindo na construção de repositórios que suportem o reuso. Algumas delas desenvolvem seus repositórios para uso interno enquanto

outras produzem-nos com propósitos comerciais. Com base nesse mercado é possível avaliar os produtos disponíveis e analisar os principais requisitos apresentados na seção anterior.

#### 2.4.1 Repositórios na Internet

Como a Internet tornou-se nos últimos tempos o maior e mais eficiente meio para o compartilhamento de informações, é esperado que torne-se também uma grande fonte de reuso para as organizações e desenvolvedores em geral. É possível encontrar por toda a rede diversos repositórios, sendo a grande maioria deles repositório de componentes de software. A seguir serão apresentados, juntamente com suas principais características, alguns repositórios encontrados de domínio público.

*ComponentSource*<sup>TM</sup> (<http://www.componentsource.com>) – ComponentSource.com é um mercado de opções destinado àqueles responsáveis por especificar, localizar e obter componentes de software. Esse mercado está aberto, através de uma política própria, à inserção dos mais diversos componentes independentemente de autores e de plataforma. O site disponibiliza a seus usuários algumas vantagens, tais como, avaliação gratuita de versões, suporte técnico, atendimento em vários idiomas, centro de soluções para construção de componentes e para comércio eletrônico. O catálogo de componentes pode ser explorado através de categorias, autores ou em ordem alfabética. A pesquisa, realizada através de palavras-chave, pode ser restringida a um determinado tipo de componente (Java, COM, Visual Basic, Delphi ou *Business*). Para cada componente existem informações técnicas de utilização, preço, compatibilidade, pré-requisitos, licença de uso, descrição, dentre outras.

*Developer.com* (<http://developer.earthweb.com/>) – A missão do *developer.com* é atender às necessidades dos profissionais de tecnologia da informação em todo o mundo. Os site apresenta material para os mais diversos tipos de tecnologia, entre eles: ActiveX, ASP, C++, Cobol, Java, JavaScript, XML e muitas outros. Permite-se a inclusão de novos recursos e a exploração de catálogos pelos tipos de tecnologia existentes. A pesquisa pode ser feita por palavras-chaves no repositório como um todo ou em algumas seções do site pré-definidas. Essas pesquisas permitem também que seja especificado um intervalo de tempo no qual deseja-se fazer a procura. O resultado obtido a partir da exploração por tipos de tecnologia é, ainda, subdividido em categorias como educação, finanças, jogos, gráficos, etc. Para cada um dos recursos encontrados há uma breve descrição do mesmo, uma referência para o autor e um link para a página do produto onde poderão ser encontradas informações mais detalhadas.

Outros repositórios disponíveis na Internet podem ser encontrados em: <http://www.flashline.com> ; <http://www.partbank.com/> ;

#### 2.4.2 Outros Repositórios

A prática do reuso possibilita a obtenção de benefícios na produtividade, qualidade, custos, tempo e esforço ao se desenvolver um sistema. Com isso, muitas organizações desejam manter os ativos por elas desenvolvidos somente para utilização interna para que possam futuramente ser beneficiadas em suas estratégias de negócio. Isso impossibilita a divulgação dos mesmos em repositórios de domínio público. Baseadas nessa necessidade é que diversas empresas estão investindo na construção de repositórios que sirvam somente aos interesses internos de uma organização. A seguir serão apresentados alguns desses produtos<sup>2</sup>.

*Microsoft Repository* – Esse repositório provê para os ambientes de desenvolvimento Microsoft um local onde é possível integrar metadados para o desenvolvimento de aplicações.

---

<sup>2</sup> As características e funcionalidades dos produtos apresentados foram obtidas através das suas páginas comerciais. Nenhum desses produtos foi instalado ou testado.

Assim como nos outros repositórios o maior objetivo dessa tecnologia é compartilhar informações que facilitem e estimulem a prática do reuso. De modo a permitir a compreensão dos ativos reusáveis por todos os seus usuários a Microsoft, juntamente com outras empresas, desenvolveu um padrão para os metadados do repositório chamado de *Open Information Model* (OIM). Para a troca de informações entre vários repositórios o formato adotado pela empresa foi o formato XML. Assim, para a comunicação entre eles, todas as informações presentes na descrição de um ativo, que estão de acordo com o padrão OIM, são inseridos em elementos pré-definidos de um documento XML. O repositório oferece ainda funcionalidades como controle de versões, controle dos relacionamento entre ativos e controle de acesso aos mesmos. Outros tipos de informações sobre o produto podem ser encontradas em <http://msdn.microsoft.com/repository/>.

*Unisys Universal Repository* - O repositório produzido pela *Unisys*, também conhecido como UREP, é um sistema de informação para definir, integrar e gerenciar metadados e informações de negócio. As principais funcionalidades oferecidas pelo repositório são: controle de versões, definição e gerenciamento de metadados, suporte a padrões como UML e ActiveX, interoperabilidade com outras ferramentas e disponibilização de serviços para navegadores de Internet (*browsers*). Maiores informações podem ser obtidas em <http://www.unisys.com/marketplace/urep>.

*Component Capitol*<sup>TM 3</sup> - O *Component Capitol*<sup>TM</sup> é um repositório desenvolvido pela *Objects Components Corporation* (OCC) para o armazenamento de componentes implementados em linguagens orientada a objetos tais como, Java, C++, Eiffel e SmallTalk. Equipes que necessitam diminuir o tempo de desenvolvimento e aumentar a qualidade de suas aplicações podem pesquisar nesse repositório para encontrar componentes que se adequem às suas necessidades. As funcionalidades disponíveis são: pesquisa e inserção de componentes, exploração dos seus catálogos e exploração de catálogos de vendedores, categorias, plataformas e tecnologias. Outras informações podem ser obtidas em <http://www.object-components.com/ComponentCapitol/>

### **3 ProReuso: Um Repositório de Componentes**

#### **3.1 Motivação**

A idéia da construção de um repositório surgiu com a necessidade de se criar um local para o armazenamento dos componentes desenvolvidos por uma empresa de software, no caso deste trabalho a Oracle do Brasil, que pudesse estar disponível via web a todos envolvidos no processo de desenvolvimento e manutenção de software.

Observou-se que a prática do reuso pode trazer grandes benefícios à uma organização. Para isso, é necessário que todos os seus membros tenham conhecimento dos componentes outrora desenvolvidos para que possam avaliá-los e, possivelmente, optar pelas sua reutilizações em novos projetos.

A principal intenção é possibilitar que no futuro todos os projetos sejam compostos de pequenos componentes diminuindo, assim, o prazo de entrega ao cliente e aumentando a produtividade e a qualidade desses projetos.

Outra intenção é permitir que o repositório sirva, também, como um banco de conhecimentos visto que um componente de software é o fruto de tarefas baseados no uso intensivo de conhecimento. Além de artefatos de software, como o código fonte dos componentes, um

---

<sup>3</sup> Atualmente existe um projeto sendo estudado pela *Objects Components Corporation* para transformar o *Component Capitol*<sup>TM</sup> num repositório para a Internet.



repositório pode também ser chamado a gerenciar outros tipo de documentos, como por exemplo, propostas, especificação de requisitos, memorando, etc., ou seja, toda a informação utilizada como insumos e como sub-produtos das diferentes atividades de desenvolvimento de software.

Para se atingir os objetivos acima citados, considerou-se fundamental que os ativos os quais constituiriam o repositório pudessem ser inspecionados seguindo regras de validação e controle de qualidade. Existiria, então, um rígido controle do que seria inserido para que não houvesse ativos desnecessários ou de baixa qualidade que pudessem desmotivar a utilização do repositório devido a falta de credibilidade das informações ali armazenadas.

Esses requisitos podem ser atendidos através da implementação de uma rotina de *workflow* permitindo a avaliação desses ativos por pessoas especializadas.

Por fim, a possibilidade de organizar as características dos ativos em formato XML permite não só uma pesquisa mais eficiente, mas possibilita, também, futuras adaptações ao repositório. Entre elas destacam-se a integração de ferramentas através da comunicação no padrão XML e a contextualização de interfaces através da definição de folhas de estilo apropriadas a cada grupo de usuários.

O conjunto de todas as características acima identificadas motivou a construção do ProReuso como sendo uma ferramenta capaz de auxiliar a organização no seu desafio de instituir a prática do reuso como uma prática cotidiana na vida dos desenvolvedores de software.

### **3.2 Descrição e Funcionalidades**

O ProReuso é um repositório desenvolvido para o ambiente web que permite o armazenamento, a pesquisa e a recuperação de componentes e de documentos a eles relacionados. Um componente é representado por um *JavaBeans*, *Enterprise JavaBeans*, documentos representam casos de uso, diagramas de interação, diagramas de classes, *JavaDoc*, apresentações, interfaces e manuais de usuário, *white papers* e outros.

A inserção dos ativos é aprovada após um processo de avaliação e validação do qual participam gerentes de projetos e especialistas em qualidade. Cada um dos participantes desse *workflow* analisa o documento segundo critérios pré-definidos e pode aprovar, rejeitar, ou aprovar com correções cada um dos ativos submetidos. O descrição desse processo será apresentada na próxima seção.

Cada documento submetido deve estar relacionado a um componente e cada componente possui uma página no site com uma referência a todos os seus documentos. Nessa página é também possível visualizar as características de todos os ativos bem como os comentários que forem inseridos pelos usuários a respeito do componente em questão. Após a submissão uma referência para o ativo já é inserida na página correspondente com uma observação indicando que o mesmo ainda não foi aprovado.

Cada ativo apresenta um conjunto de características que são agrupadas num documento XML. Esse documento servirá como base para a rotina de pesquisa e recuperação de ativos do ProReuso.

Para a pesquisa o usuário pode especificar em quais campos do documento XML ele deseja que ela seja efetuada. Assim, o usuário pode eliminar informações desnecessárias e mais facilmente localizar os ativos desejados.

O ProReuso pode ser acessado por usuários públicos (dentro da intranet) ou por usuários cadastrados através do fornecimento de login e senha. Sendo assim, o sistema oferece um controle de segurança da seguinte forma: Apenas os usuários cadastrados podem submeter

componentes e documentos, visualizá-los e recuperá-los. Os usuários públicos podem fazer pesquisas, mas não podem recuperar os ativos encontrados. A eles só é permitido o acesso a documentos informativos tais como *readme*, *white papers* e manuais do usuário.

O repositório disponibiliza, ainda, um catálogo com todos os componentes disponíveis. Essa catálogo traz a descrição dos mesmos e uma referência à sua página para os usuários cadastrados.

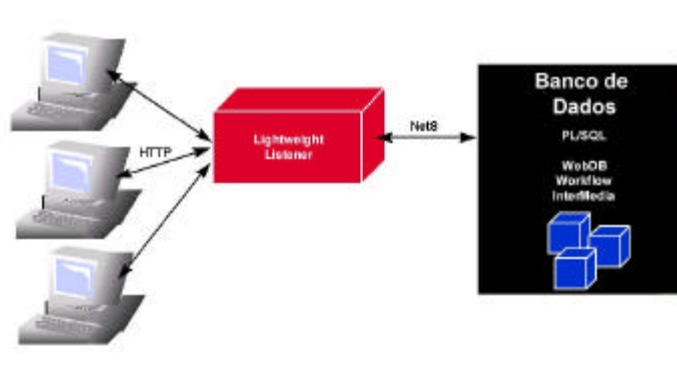
Por fim, através do ProReuso, é possível compartilhar experiências através da inserção de comentários. Um usuário que já tenha utilizado um componente pode inserir dados relevantes sobre sua experiência com o mesmo e esses dados estarão disponíveis a todos os outros usuários para que eles sigam as suas recomendações.

As principais funcionalidades disponíveis aos usuários do ProReuso são:

- Submissão de Componentes e Documentos – responsável por iniciar o processo de aprovação de ativos. Na submissão, o usuário fornece as principais características do componente ou do documento as quais irão compor o documento XML.
- Acompanhamento de Processos – possibilita a um usuário acompanhar graficamente todos os processos por ele iniciados. O usuário pode a qualquer momento identificar em que ponto do processo se encontra o ativo submetido.
- Pesquisa – permite a procura pelos componentes e documentos que compõe o repositório. Essa pesquisa pode ser contextualizada, ou seja, é permitido ao usuário que identifique em quais campos do documento XML ele deseja que a pesquisa seja feita.
- Exploração do Catálogo – mostra ao usuário todos os componentes que fazem parte do repositório juntamente com suas descrições.
- Inserção de Comentários – permite que uma pessoa que já tenha utilizado um componente ou um documento insira comentários sobre aquele ativo de modo a compartilhar experiências.

### 3.3 Arquitetura

Para a construção do ProReuso foi adotada uma arquitetura que baseia-se, principalmente, na estrutura de um banco de dados relacional e íntegra, a essa estrutura, 3 ferramentas conforme mostra a Figura 1.



**Figura 1 - Arquitetura do ProReuso**

Como se pode observar, o ProReuso corresponde a uma instância de banco que é acessada por estações clientes através de um *listener* de interface para web. Toda a sua lógica está armazenada no banco na forma de procedimentos PL/SQL, que geram páginas HTML para serem visualizadas através de um navegador nas estações clientes. O acesso a esses procedimentos é feito através de um servidor web representado por um *listener* HTTP agrupado a um *gateway* PL/SQL

Os elementos presentes nessa arquitetura são:

- Oracle WebDB - Ferramenta utilizada para o desenvolvimento de aplicações para a Internet. Disponibiliza aos desenvolvedores funções como monitoração de aplicações e controle de acesso. Encontra-se armazenada no banco de dados e todas as suas funcionalidades são representadas por pacotes PL/SQL.
- Oracle WorkFlow - Ferramenta utilizada para a definição e automação de processos. Assim como o WebDB, encontra-se completamente armazenada no banco. É composta por um mecanismo gerenciador que controla a execução do processo e por um sistema de notificação responsável pelo envio de mensagens e pelo processamento de suas respostas.
- Oracle Intermedia - Ferramenta utilizada para a pesquisa de documentos armazenadas no banco.
- *Listener* e *PL/SQL gateway* - Esses elementos fazem parte do WebDB e atuam como a camada intermediária entre o protocolo HTTP utilizado pelo navegador e o protocolo NET8 utilizado pelo banco. Isso permite que links no site sejam requisições de procedimentos PL/SQL.

### 3.4 Processos

Nessa seção, será explicada a rotina de *workflow* modelada para o processo de aprovação de um componente<sup>4</sup>. Todo componente submetido ao repositório passará por esse processo, que corresponde a uma crivo de gerentes especializados, que se encontram numa hierarquia, e de uma grupo de qualidade, que avalia a corretude do componente em relação a padrões pré-definidos.

#### 3.4.1 Processo Aprovação de Componentes

Este processo apresenta em nível mais alto todos os passos definidos para a aprovação de um componente (ver Figura 2). O processo se inicia quando o requisitante submete um componente no repositório. Os passos do processo são:

1. Verifica-se se o requisitante corresponde ao último nível da hierarquia (f<sup>5</sup>: Verifica Nível).
2. Se for o último, o fluxo seguirá diretamente para o subprocesso de análise do grupo de qualidade (p: Verifica Qualidade- segue passo 11).
3. Se estiver em nível intermediário, verificar-se-á na hierarquia quem é o superior imediato (f: Verifica Superior).
4. Inicia o subprocesso de notificação desse superior quanto à submissão do componente, que requer sua aprovação (p<sup>6</sup>: Notifica Superior) e o requisitante é notificado de que o seu pedido foi enviado para esse superior.
5. Aguarda-se o retorno do processo anterior, que corresponde à resposta do superior, podendo esse aprovar, aprovar com maiores ou menores correções ou rejeitar o componente.
6. Caso o pedido seja aprovado, será verificado qual é o nível na hierarquia do superior avaliador (f: Verifica Nível) e se ele corresponder ao último nível, então o requisitante será notificado quanto à aprovação e dará início ao subprocesso de notificação do grupo de qualidade quanto à necessidade de avaliação do componente submetido (p: Verifica

---

<sup>4</sup> No contexto desse processo, a palavra componente estará representando tanto um componente como um documento submetido ao repositório.

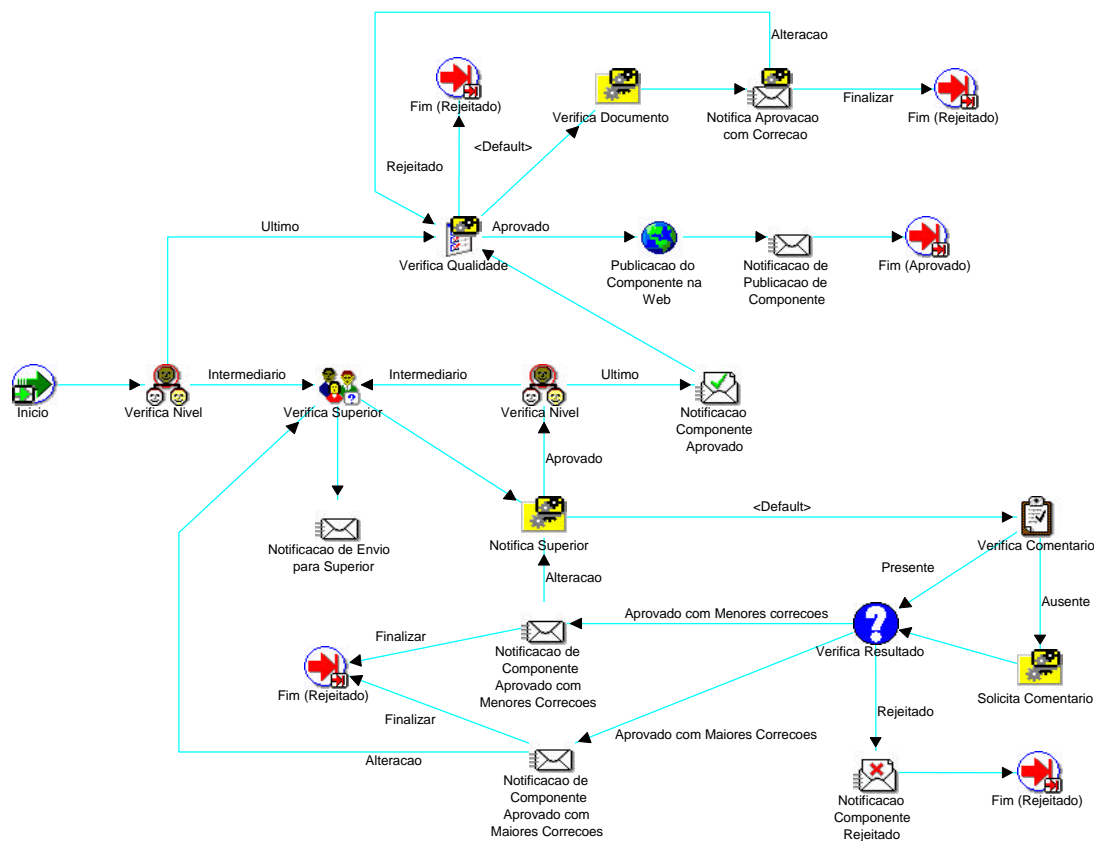
<sup>5</sup> Representa uma função definida no processo.

<sup>6</sup> Representa um subprocesso do processo principal.

Qualidade- segue passo 11). Se o superior estiver em um nível intermediário na hierarquia, será verificado quem é o superior imediato na hierarquia e ,em seguida, inicia o subprocesso de notificação desse superior quanto à submissão do componente, que requer sua aprovação (p: Notifica Superior).

7. Nos casos em que o pedido for aprovado com correções ou rejeitado, será cobrado do superior avaliador a inserção de comentários, justificando o resultado de sua avaliação. Verifica-se se comentários foram inseridos e em caso negativo, inicia-se o subprocesso que notifica o superior quanto à necessidade de inserir comentários (p: Solicita Comentário). Quando o superior inserir os comentários ou se já os tiver inserido anteriormente, verifica-se o resultado (f: Verifica Resultado) e notifica o requisitante do resultado da aprovação.
8. Caso o pedido seja aprovado com maiores correções, o requisitante terá a opção de alterar o componente e o re-submeter ao seu superior imediato. O requisitante também terá a opção de finalizar o processo caso não queira continuar. Se fizer essa opção, o processo é finalizado.
9. Caso o pedido seja aprovado com menores correções, o requisitante terá a opção de alterar o componente e o re-submeter ao último superior que analisou o componente. O requisitante também terá a opção de finalizar o processo caso não queira continuar. Se fizer essa opção, o processo é finalizado.
10. Caso o pedido seja rejeitado, o requisitante é notificado e o processo encerrado.
11. No caso de aprovação do último superior da hierarquia , será iniciado o subprocesso que solicita ao grupo de qualidade a avaliação do componente (p: Verifica Qualidade). O grupo poderá aprovar, aprovar com correções ou rejeitar o componente.
12. Caso o componente seja aprovado pelo grupo de qualidade, o componente é publicado no site do repositório (f: Publicação de Componente na Web), o requisitante notificado da publicação e o processo encerrado.
13. Caso o componente seja rejeitado, o requisitante é notificado e o processo encerrado.
14. Caso o componente seja aprovado com correções, será cobrado do analisador que ele informe as correções necessárias. Verifica-se se ele realizou essa tarefa e em caso negativo, a sua efetivação é cobrada (p: Verifica Documento). O processo só seguirá quando as correções forem inseridas. Quando isso ocorrer, o requisitante será notificado do resultado e terá a opção de corrigir o componente ou desistir do pedido, finalizando o processo. Caso escolha continuar, após corrigir o componente e o re-submeter, o subprocesso de notificação do grupo de qualidade solicitando a avaliação do componente é reiniciado (p: Verifica Qualidade).

As notificações tanto para o requisitante como para os superiores e analisadores do grupo de qualidade são feitas via e-mail. Naquelas que exigem resposta, a própria mensagem possui uma área reservada para tal.



**Figura 2 - Processo de Aprovação de Componentes Processo Notifica Superior**

Subprocesso do aprovação de componentes que corresponde à notificação de um superior quanto à submissão de um componente que requer sua aprovação. Os passos correspondentes a este sub-processo são descritos abaixo:

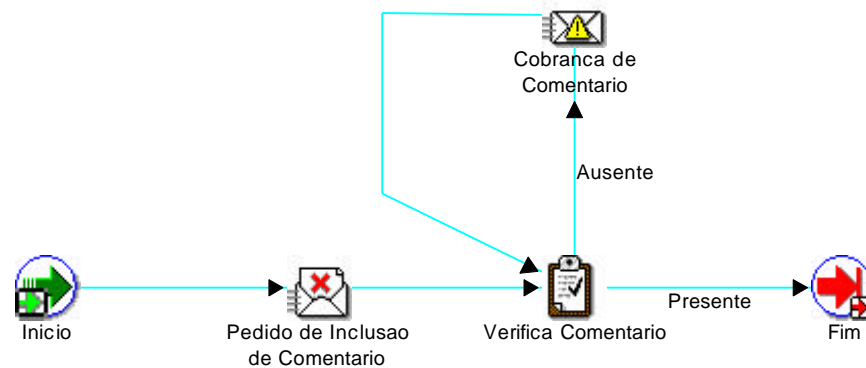
1. O superior recebe a notificação com os dados da submissão: nome e tipo do componente, nome do requisitante e o link para o arquivo do componente. No caso da submissão ser de um documento, o nome do componente a que esse documento está associado também é fornecido.
2. Na própria notificação, o superior responde ao pedido de aprovação com o resultado da sua avaliação. O resultado pode ser: aprovado, aprovado com maiores correções, aprovado com menores correções e rejeitado.
3. Quando o superior responder, o processo será encerrado.

### 3.4.2 Processo Solicita Comentário

Subprocesso do aprovação de componentes que corresponde à solicitação de inserção de comentários por parte do superior, quando esse rejeitar ou aprovar com correções um pedido de aprovação de componente (ver Figura 3). Antes já foi verificado que ele não havia fornecido nenhum comentário (f: Verifica Comentário). Os passos do processo são:

1. O superior recebe a notificação, lembrando-o do resultado por ele fornecido para determinado componente e quanto à necessidade de se fornecer comentários, justificando resultados desse tipo.
2. O superior responderá a notificação inserindo no lugar reservado para tal o comentário.
3. Verifica-se se o comentário foi fornecido (f: Verifica Comentário).

4. Caso não tenha fornecido, será enviada nova cobrança e continuará aguardando a inserção.
5. Quando a inserção tiver realmente ocorrido, o processo será encerrado.

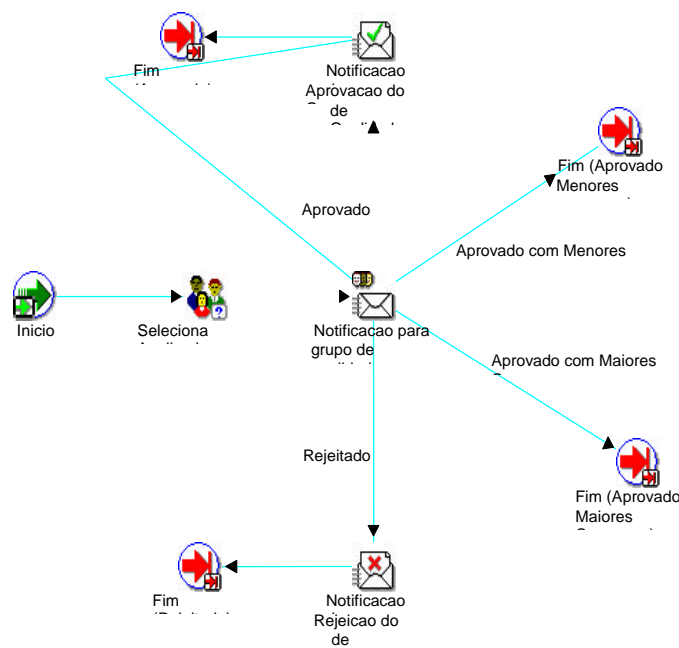


**Figura 3 – Processo Solicita Comentário**

### 3.4.3 Processo Verifica Qualidade

Subprocesso do aprovação de componentes que corresponde ao processo de análise da qualidade do componente, depois dele ter sido aprovado pelo(s) superior(s) (ver Figura 4).

1. É verificado qual o analisador corresponde ao tipo do componente que está sendo submetido, pois para cada tipo de componente um determinado membro do grupo de qualidade é responsável pela análise (f: Seleciona Analisador).
2. Esse analisador recebe a notificação, solicitando a análise da qualidade para o componente e alertando da necessidade de se inserir um documento de correções caso se aprove com correções.
3. Na notificação são fornecidos o nome e tipo do componente, o requisitante e o link para o arquivo do componente. Caso tenha sido feita a submissão de um documento, também será fornecido o nome do componente ao qual o documento está associado.
4. Na notificação, ainda será lembrado ao analisador que para resultados do tipo aprovado com correções, faz-se necessário a inserção de um documento com as correções que devem ser feitas. Na mensagem da notificação, existe um link para o formulário onde esse documento pode ser submetido.
5. Na própria notificação, o analisador responde ao pedido com o resultado da sua análise. O resultado pode ser: aprovado, aprovado com maiores correções, aprovado com menores correções e rejeitado.
6. Quando o analisador responder, o processo será encerrado.



**Figura 4 – Processo Verifica Qualidade**

### 3.4.4 Processo Verifica Documento

Subprocesso do aprovação de componentes que corresponde ao processo de verificação e cobrança da inserção de um documento de correção por parte do analisador do grupo de qualidade, quando este aprova um componente com correções.

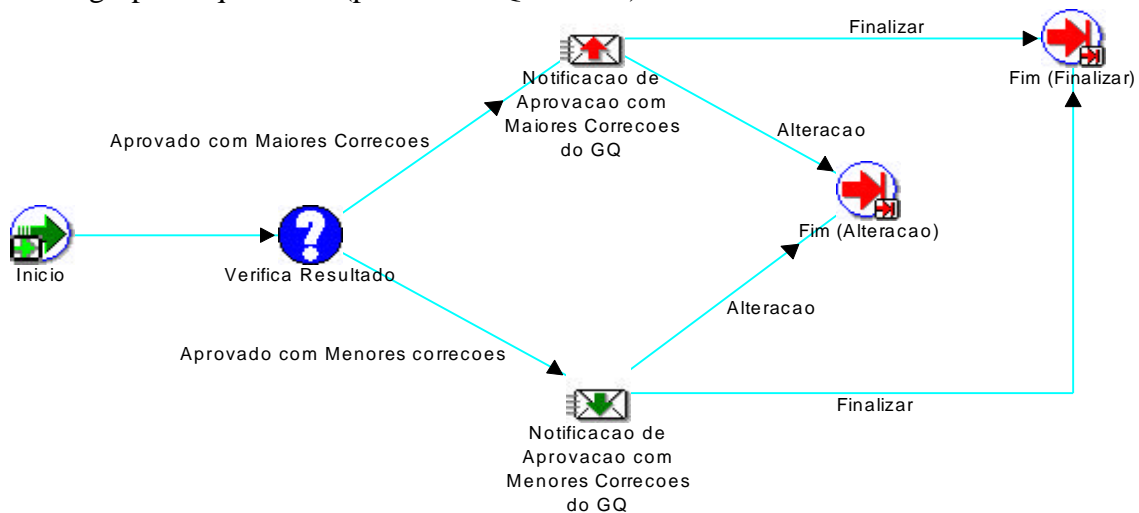
- 1) É verificado se o documento de correções foi inserido.
- 2) Em caso afirmativo, o processo será encerrado.
- 3) Em caso negativo, o analisador recebe a notificação, lembrando-o da necessidade de inserir um documento de correção quando se aprova com correções e solicita que o faça, devido ao resultado fornecido por ele para o componente em questão.
- 4) Na notificação são fornecidos o nome e tipo do componente, o requisitante, o link para o arquivo do componente e o link para o formulário de inserção do documento de correções. No caso de submissão de documento, será fornecido também o nome do componente ao qual ele está associado.
- 5) Na própria notificação, o analisador responde à notificação, após inserir o documento através do formulário indicado, informando que o documento já foi inserido.
- 6) Quando o analisador responde, será verificado se o documento foi realmente inserido e volta-se ao passo 1 do processo.

### 3.4.5 Processo Notifica Aprovação com Correção

Subprocesso do aprovação de componentes que corresponde ao processo de verificação com qual tipo de correções, maiores ou menores, o componente foi aprovado pelo analisador do grupo de qualidade e a notificação do requisitante quanto a esse resultado (ver Figura 5). Os passos desse subprocesso são descritos abaixo:

1. Verifica-se qual tipo de resultado foi dado ao componente, aprovação com maiores ou menores correções.

2. O requisitante é notificado informando qual foi o tipo do resultado da análise do grupo de qualidade. Na mensagem será fornecido o link para o arquivo do documento de correções submetido pelo analisador.
3. Em qualquer dos casos, maiores ou menores correções, o requisitante poderá optar por alterar o componente e re-submetê-lo ao grupo de qualidade, ou por desistir da submissão, finalizando o processo. No caso de escolher alterar, após responder à notificação será reiniciado o processo de solicitação de análise de componente ao grupo de qualidade (p: Verifica Qualidade).



**Figura 5 - Processo Notifica Aprovação com Correção**

#### 4 Conclusões

O mercado cada vez mais competitivo em que as empresas desenvolvedoras de software se encontram atualmente exige dessas um comportamento bastante diferente daquele praticado durante muitos anos. Softwares grandes e complexos que apresentam vários defeitos e que não satisfazem às necessidades dos usuários não são mais aceitos por seus clientes. Devido a essas exigências, as empresas que se preocupam em sobreviverem no mercado buscam alcançar objetivos que exigem bastantes esforços e requerem tempo para se realizarem, tais como: aumento de qualidade, satisfação do cliente, melhoria técnica, redução de prazos de entrega, aumento da produtividade e redução de defeitos e de riscos. Além disso, com o crescente avanço da tecnologia, essas empresas precisam se adaptar para melhor atender a todas as expectativas de seus clientes.

Para atender a essas necessidades o reuso propõe o desenvolvimento de sistemas a partir de produtos de software preexistentes, de modo que semelhanças entre requisitos e arquiteturas de diversos sistemas possam ser melhor exploradas. Com a identificação das semelhanças é possível se reduzir o tempo de entrega e o custo de desenvolvimento e aumentar a produtividade e qualidade do produto final.

De modo a potencializar a prática do reuso nas organizações, muitas empresas têm feito uso de um repositório de onde é possível recuperar ativos pré-construídos que poderão simplificar o processo de desenvolvimento e auxiliar na disseminação de experiências e práticas comuns no ambiente corporativo. Sendo assim, um repositório é um fator técnico que torna possível uma estratégia de reuso adotada por uma organização. Entretanto, outros fatores como o gerenciamento, o incentivo à utilização e o treinamento de uso de repositório são também fundamentais para possibilitar essa estratégia de negócio.



A implementação do ProReuso levou em consideração fatores que podem ser determinantes para o êxito no processo de reuso. Entre eles destacam-se: o processo de análise, validação e aprovação de componentes que possibilita garantir a qualidade do que está sendo inserido no repositório; a pesquisa contextualizada que permite uma recuperação mais apurada dos ativos e a disponibilização em ambiente web a qual possibilita o acesso remoto a todos os funcionários da empresa.

Tendo em vista a intenção inicial da empresa na construção de um repositório de componentes executáveis, o ProReuso foi focado no armazenamento desses componentes e dos documentos relacionados a cada um deles. Entretanto, observou-se que é possível e interessante ampliá-lo para um repositório de conhecimentos gerais onde seria permitido a inserção não só de ativos de software, como também de procedimentos inerentes a um determinado processo, regras da organização para a execução de atividades, materiais para treinamentos e tudo o mais que representar o patrimônio informativo de uma empresa. Isso é o que propõe os pesquisadores do Gerenciamento do Conhecimento (*Knowledge Management*). Outro fator a ser considerado como uma futura adaptação é permitir que os próprios autores dos ativos possam modificar as características dos mesmos devido às constantes mudanças presenciadas pela área tecnológica.

Para finalizar, destaca-se que a implementação do ProReuso mostrou ser viável a construção de um repositório por aquelas organizações que desejam alcançar os benefícios obtidos com o reuso. Isso possibilitará que as pequenas empresas também participem do grupo de grandes empresas que se já se beneficiaram dessa prática, e que caminhem, junto à Engenharia de Software, rumo à excelência no desenvolvimento de sistemas.

### **Referências Bibliográficas**

- [BADCOCK 1998] BADCOCK, Charles. Repositories Reap E-Commerce Role. Disponível na Internet. <http://www.zdnet.com/flitters/printfriendly/0,6061,2149796-35,00.html>, 1998
- [BASILI et al., 1996] BASILI, Victor R., et al.. How Reuse Influences Productivity in Object-Oriented Systems. In: Communication of the ACM No 10, outubro de 1996. v. 39.
- [BOOCH et. al., 1999] BOOCH, Grady, et al.. The Unified Modeling Language User Guide. 1.ed.Massachusetts: Addison-Wesley, 1999.512p
- [EZRAN et al. 1999] EZRAN, Michel, et al.. Practical Software Reuse: The Essential guide. 1999.185 p
- [FAVARO et al. 1998] FAVARO, John M., et al.. Value Based Software Reuse Investment. In: Annals of Software Engineering, 1998. v.5, p.5-52
- [GRAHAM 1999] GRAHAM, Ian, QUIN Liam XML Specification Guide [s.l]: Willey Computer Publishing,1999. 431p.
- [KRUEGER 1992] KRUEGER, Charles W. Software reuse. In: ACM Computing Surveys. No 2, junho 1992. v. 24, p.131-183.
- [HENNINGER 1997]HENNINGER, Scott. An Evolutionary Approach to Constructing Effective Software Reuse Repositories. In: : ACM Transactions on Software Engineering and Methodology No 2, abril 1997. v. 6, p.111-140.
- [POULIN 1998] POULIN, Jeffrey S.. The Foundation of Reuse. Disponível na Internet. <http://www.umcs.maine.edu/~ftp/wisr/wisr9/final-papers/PoulinJ.html>, 1998
- [UREP 2000] UNISYS. Unisys Universal Repository. Disponível na Internet. <http://www.unisys.com/marketplace/urep>, 2000