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@yahoo.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 Introduction

Motivation

Reuse proposes that software development to be done from preexisting software products, so that similarities among requisites and architectures of different systems can be better explored [FAVARO et al., 1998] [JACOBSON et al 1997] [BASILI et al., 1996]. The identification of such similarities may result, among other things, into significant cost and time reductions, or else, into productivity for the developer enterprises. Moreover, to each product use, its efficiency and validity are checked, consequently guaranteeing a greater quality and reduction of mistakes in final products.

At its turn, a component repository represents a potential tool for the exercise of reuse inside organizations. When the components stored in it are well organized and managed, this potential becomes a reality. Through a repository sustaining the reuse politics inside the enterprise, aiming at the benefits of this practice, there will be an increase in the levels of productivity and quality in the process of development of the organization using it. This means that software reuse is a business strategy that should be sustained, in a technological point of view, by a corporative components repository.

Therefore, from these considerations and taking into account the increasingly more competitive market, in which software enterprises are inserted, the implementation of a software components repository to enable the reuse process becomes a differential factor that allows them to enter the group of those that have access to the benefits brought by reuse.

1.1 Objective

The objective of this article is to describe the experience of a components repository directed by a reuse process.

At this project it is defined that reuse is a systematic practice of developing new applications from pre-constructed software actives that present similarities of requisites and/or architectures with what is being developed [KRUEGER, 1992]. Since it is a systematic practice, it requires discipline and involvement from all enterprise members. Moreover, it requires a process of internal awareness, so that everyone understands the potential for productivity increase, products quality, and economy related to the costs and time of development that this practice represents.

In the definition of reuse, we have used the word software active. Software active is defined as the item that can be reused, consisting of any reusable software product, of any development phase [EZRAN et al., 1999]. This was chosen due to the fact that the term brought the idea of value and investment to the enterprise with it. Since these ideas correspond to the concept of reusable software, we plan to make clear in this project that the term active is well employed.

Still in relation to the term active, it is important to explain that in the repository implementation phase, this term was substituted by software component. Such substitution does not mean that what was said in the previous paragraph was forgotten or contradicted. It happened because the enterprise that motivated the construction of this repository, Oracle of Brazil, had the initial intention of building a repository focused in executable components and documents related to them, such as the user's manual, UML diagrams [BOOCH et al., 1999], technical presentations, etc., that would attend mainly to its development area. We can also mention another less relevant factor, which is the fact that the term components is more familiar to people in general when it refers to software reuse. Therefore, despite the substitution, we can still think in software actives being stored in the repository.

Thus, ProReuse is a reusable components repository consisting of a web application made available to Oracle of Brazil's employees through the enterprise's Intranet.

With ProReuse, the employee may both feed the repository with new components and/or documents related to them and locate and recuperate others already made available. To allow the feeding, ProReuse provides the components and documents submission functionalities, which enables the entrance of characteristics from them so they can be stored inside a defined organization.

In order to implement a managing process of what is being made available in the repository, a components[1] approval process is implemented in the ProReuse through a workflow. This process represents the evaluation done by project managers and quality specialists to which every component or document submitted to the repository has to pass. Then, we have a guarantee that everything made available has already been checked and approved by specialists.

For the efficiency in the research of actives made available in the ProReuse, resources of textual structuration offered by the XML technology (*eXtensible Markup Language*) [GRAHAM, 1999] were used. Each active has an associated XML document fed during submission, which stores information on it in a structured way, corresponding to the active characteristics. The indexation of these documents is done through a tool (Oracle Intermedia) made available by the database, using the structuration done according to the facilities provided by the XML.

Thus, we consider these two last important differential functionalities made available in ProReuse. This way, we believe that the ProReuse represents an example of repository that serves, in fact, as an auxiliary tool for the practice of reuse inside an organization.

2 Repository: Components Reuse

2.1 Introduction

With the growing importance attributed to software and information reuse, there have been many discussions about the organization and centralization of these values. The repository comes up as a solution to solve a paradox observed in the organizations: we know that the sharing of experiences and information may bring many competitive advantages. However, many enterprises are barred to seize such benefits because they cannot deal with the variety and the inefficient distribution of desired information, and, often, the do not even know about the available information [HENNINGER 1997].

The term repository may be used for different purposes. Among them, we highlight:

1. The storage of models – a development process involves the generation of many artifacts that are normally stored in different places. A repository allows the storage centralization of these artifacts and the control of each one of their versions.
2. Integration of tools – the models generated during a development process often originate from distinct tools. Therefore, for them to exchange information, it is necessary to have a common "language" through which they can communicate. The repository may serve as a temporary storage and the intermediation between the tools, that is, it may transform the entrance model of one of them into a model compatible to the other.

[1] In this process, the term "component" is also used to refer to documents, in order to be coherent to the application's name.

3. Maintenance of Legated Systems – a repository can be useful also for the storage of data related to old systems. These data may afterwards be analyzed to determine the possible impacts generated by modifications.

After analyzing such functionalities, we can observe the repository is used for the storage of metadata. This restriction, as stressed in [BABCOCK, 1998] is what differentiates them from the databases.

For the purpose of this project, a repository will be considered as the place where software actives will be stored and classified in a way as to facilitate their correct use and recuperation.

2.2 Characteristics of a Repository

The main functionalities of a repository are the researches and the recuperation of software actives. However, the presence of some other functionality may facilitate, and, with it, stimulate its use. According to what is proposed by [EZRAN et al., 1999], we will present, next, some of the main functions a repository may provide to its users. It is worth to stress that these functions are not obligatory and vary according to the context in which they are inserted and to the needs of the organization when using a repository.

Identification and description: in order to describe an active, it is possible that a set of characteristics are attributed to it, such as name, domain, key words, among others that identify and differentiate them from the other actives that compose this same repository. To each one of the actives stored, it should be possible to identify it homogeneously inside a repository, that is, the same type of actives present the same set of characteristics. Nevertheless, this does not mean that different actives should have the same values for this set of characteristics.

Insertion: a repository should allow the authorized users to insert new actives or, still, new versions of them. Insertion means to add to the repository the body and the description of the active, according to what was presented in chapter two.

Catalog Exploration: users of a repository should be allowed to explore the active catalogs in order to know and analyze the available actives' characteristics.

Textual research: a repository should allow its users to perform more specific researches in the actives description. As a research result, one or more actives that attend the desired conditions will be obtained. After observing the details, it is possible to choose between a greater specification and generalization of the previous criteria.

Recuperation: after the identification of the desired active, a repository should allow its users to recuperate this active so they can use it afterwards in a reuse process.

Organization and research: as seen before, the catalog exploration functionality is not enough as the number of available actives increases. Moreover, textual research often demands a lot of time. Due to these factors, some repositories may adopt new ways of organizing the actives' characteristics in a way as to allow the research in the repository to be based in other criteria.

Record: It is important for the managing of a repository the storage of information of use, modifications, creation, and exclusion of each one of the available actives. These pieces of information should compose a historic basis that will facilitate their analysis and reuse.

Measurement: a repository may collect statistics that facilitate its management. Some of the main statistics that can be adopted are: access frequency to the repository, number of available actives, recuperation rate, well-succeeded researches percentage, frequency with which a certain active is analyzed or modified, among others.

Access control: a repository may adopt a security politics so that certain functionalities are accessible only to authorized people. For instance, a security politics, specific to an enterprise, where textual research is available to all employees may be defined, but the actives recuperation would only be available to the employees of the development area.

Management of versions: A repository may contain many versions of a same active and, thus, the creation of a mechanism to control these versions and establish a relationship between them is recommended.

Modifications Control: the provision of some functionalities to perform the managing of actives modifications in a repository is recommended. These functionalities include proceedings for the request of alterations, discussions, and permission for them. It is extremely important for the maintenance of a repository's consistency that any intentions to modify the actives are communicated to that responsible for their administration. Him, together with other responsible people, may analyze the requested alterations and maintain the coherence between the different software actives.

Changes notification: it is possible that, at any moment, an active is modified or even the repository itself is submitted to some alterations in its main functionalities. Then, a repository may provide functions to notify their users about the modifications recently done, such as the insertion or exclusion of actives, alterations in documents, availability of new functionalities, new security politics, among others.

After the main functionalities of a repository are presented, it is necessary to analyze some of the non-functional aspects that characterize them. These aspects may also, if correctly used, bring more incentives to the repository's use.

Quantity: the repository may be unique or there may be a set of them [EZRAN et al., 1999]. The unique one is indicated for small organizations where there is not a great departmental division and, thus, the kinds of actives are not distinct and are not understood by all of its users. For the greater organizations or geographically distributed, we suggest the use of many repositories, each one related to an action area of the enterprise. In this case, it is important to have a centralized and efficient management in order to prevent inconsistencies in the duplications that exist between some repositories. Those that consider the information unnecessary for a certain set of users defend the use of many of them, but they are really important for another one, prejudice and discourage the practice of reuse in the organizations.

Access through the net: as the great majority of enterprises are migrating their computational models to centralized models, a typical requisite is that the repository may be accessed at any point of a net [EZRAN et al., 1999]. This is particularly important when its users are geographically distributed. What has normally been observed in repositories implementation is the use of client-server models in 3 (three) or more levels. In the models of three levels, one is the responsible for the storage of actives, the second for the processing of the main system's functionalities, such as research and recuperation, and the third one corresponds to client application through where the repository may be accessed. The number of levels and the way of implementing them may vary among the different organizations.

2.3 Management

The management of actives means to discover, identify, and manage each active as a reuse opportunity [UREP, 2000]. According to what has been previously presented, many enterprises have information scattered throughout the entire organization. It is up to the manager of a repository to identify those that aggregate values to the organization and include them in this repository so they can be shared by everybody.

As the repository represents a collection of actives, it is necessary to have a managing process in order to better control the quantity and quality of available actives, as well as their locations and main characteristics. Adding to this, in the case of non-centralized repositories, it is important to have a thorough control in order to prevent inconsistencies between the many existing repositories.

The actives catalogs control is a vital activity in the managing of a repository. In it, there can be found references to each one of the actives that compose it and that will be fundamental for the practice of systematized reuse. Its absence may be the greater contribution for the chaos. [EZRAN et al., 1999].

As could be observed, the real value of a repository lies in the capability of finding what is being searched or what attends the user needs. Therefore, the managing should contribute to keep information coherent and adequate to a certain group of users. This includes, as shown in the previous section, control of versions, of modifications, and adequate classification of each one of the actives available in a repository.

Another important point to be observed in the managing of a repository is the quality control that should be imposed to the actives that constitute it. Finally, the fact that the actives are available and able to be found is not enough. It is also necessary that they can be understood so then they can be reused. At this point we highlight the importance of patterns definition for the actives and the managers' role in the validation of these patterns.

2.4 Repositories Market

With the growing motivation for the reuse of software actives in the development processes, many enterprises are investing in the construction of repositories that can bare reuse. Some of them develop their repositories for internal use while others produce them for commercial purposes. Based on this marked it is possible to evaluate available products and analyse the main requisites presented in the previous section.

2.4.1 Repositories in the Internet

As the Internet has recently become the greatest and the most efficient mean to share information, it is expected that it also becomes an extraordinary source of reuse for the organizations and developers in general. It is possible to find many repositories throughout the net; the great majority of them are software components repositories. Next, we will present, together with their main characteristics, some repositories found in the public domain.

ComponentSource™ (http://www.componentsource.com) – ComponentSource.com is an options market made for those responsible for specifying, finding, and obtaining software components. This market is open, through personal politics, to insertion of the most diverse components independently from the authors and platform. The site provides some advantages to its users, such as the free evaluation of versions, technical support, service in many idioms, a solutions center for the construction of components and electronic commerce. The components catalog may be explored through categories, authors or in alphabetical order. The research, done through keywords, may be restricted to a certain type of component (Java, COM, Visual Basic, Delphi or *Business*). For each component there are use technical information, price, compatibility, prerequisites, use license, description, among others.

Developer.com (http://developer.earthweb.com/) – Its mission is to attend the needs of the information technology professionals throughout the world. The site offers material for the most different types of technology, which includes: ActiveX, ASP, C++, Cobol, Java, JavaScript, XML, and many others. It is allowed to include new resources and to explore the catalogs from the types of existing technology. The research may be done by keywords in the

entire repository or in some predefined sections of the site. These researches also allow us to determine a time interval in which we intend to perform the search. The result obtained from the exploration by types of technology is, still, subdivided into categories as education, finances, games, graphs, etc. For each one of the resources found there is a brief description of it, a reference for the author, and a link for the product's page where more detailed information may be found.

Other repositories available in the Internet may be found in: http://www.flashline.com; http://www.partbank.com/;

2.4.2 Other Repositories

The reuse practice enables users to obtain benefits in productivity, quality, costs, time, and effort to develop a system. Due to this, many organizations wish to keep their actives developed by them only for internal use so they can be beneficiated in the future in their business strategies. This bars the divulgation of these actives in public domain repositories. Based on this need, many enterprises are investing in repositories that serve only to an organization's internal interests. Next, some of these products[2] will be presented.

Microsoft Repository – This repository provides for the Microsoft development environments a place where it is possible to integrate metadata for the development of applications. As in other repositories, the greatest objective of this technology is to share information that would facilitate and stimulate reuse practice. In order to make the reusable actives comprehensible to all its users, Microsoft, together with other enterprises, has developed a pattern for the repository metadata, called Open Information Model (OIM). For the information exchange between many repositories, the format adopted by the enterprise was XML. Thus, for the communication between them, all information present in the description of an active that are in accordance with the OIM pattern are inserted in predefined elements of an XML document. Also, the repository offers functionalities as control of versions, control of the relationship between actives and control of access to these actives. Other types of information on the product may be found in http://msdn.microsoft.com/repository/.

Unisys Universal Repository – The repository produced by *Unisys*, also known as UREP, is a system of information to define, integrate, and manage metadata and business information. The main functionalities offered by this repository are: control of versions, definition and managing of metadata, support to patterns as UML and Active X, interoperability with other tools and availability of services for Internet navigators (*browsers*). For more information, access http://www.unisys.com/marketplace/urep.

Component Capitol™ [3] – The *Component Capitol*™ is a repository developed by the *Objects Components Corporation* (OCC) for the storage of components implemented in objects oriented languages such as Java, C++, Eiffeil e SmallTalk. Staffs that need to decrease development time and increase the quality of its applications may research in this repository to find components that are adequate to their needs. The available functionalities are: research and insertion of components, exploration of its catalogs and exploration of catalogs of sellers, categories, platforms, and technologies. Other information may be obtained in http://www.object-components.com/ComponentCapitol/

[2] The characteristics and functionalities of the products presented were obtained through its commercial pages. None of these products was installed or tested.

[3] Nowadays there is a project being studied by the *Objects Components Corporation* to transform the *Component Capitol*™ into a repository for the Internet.

3 ProReuse: A Components Repository

3.1 Motivation

The construction of a repository idea came up with the need to create a place for the storage of components developed by a software enterprise, in this case the Oracle of Brazil, that could be available to all the involved in software maintenance and development process.

It was observed that the practice of reuse could bring many benefits to an organization. For that, it is necessary that all of its members know the components developed so they can evaluate them and, possibly, choose for their reuse in other projects.

The main intention is to enable the projects to be composed of small components in the future. Therefore, this would decrease the time for delivering the product to the client and increasing the productivity and quality of these projects

Another intention is to allow the repository to serve, also, as a knowledge base, since a software component is the product of tasks based on the intensive use of knowledge. Beyond software artifacts, as the components font code, a repository may also be called to manage other types of documents, such as proposals, requisites specification, memorandum, etc., that is, all the information used as input and as sub product of the different software development activities.

To reach the objectives cited before, it was fundamental the actives that would constitute the repository were inspected according to validation and quality control rules. Therefore, there would be a thorough control of what would be inserted so there would not be unnecessary or low quality actives that could discourage the repository use due to the lack of credibility of the information stored there.

These requisites may be attended through the implementation of a *workflow* routine, allowing the evaluation of these actives by specialized personnel.

At last, the possibility of organizing the XML format actives characteristics allows not only a more efficient research, but it also enables future adaptations to the repository. Among them we highlight the integration of tools through the communication in the XML pattern and the interfaces contextualization through the definition of style sheets appropriate to each user group.

The set of all the characteristics identified above has motivated the construction of ProReuse as being a tool capable of helping the organization in its defiance point of instituting the performance of reuse as a day to day practice in the life of software developers.

3.2 Description and Functionalities

ProReuse is a repository developed for the web environment. It allows the storage, research and recuperation of components and documents related to them. A component is represented by a *JavaBeans*, *Enterprise JavaBeans*, documents that represent use cases, interaction diagrams, classes diagrams, *JavaDoc*, presentation, interfaces and user manuals, white papers and others.

The actives insertion is approved after an evaluation and validation process in which project managers and quality specialists participate. Each one of the participants in this workflow analyze the document according to predefined criteria and may approve, reject, or approve with corrections each one of the actives submitted. The description of this process will be presented in the next section.

Each document submitted should be related to a component and each component has a page in the site with a reference to all of its documents. In this page is also possible to visualize the characteristics of all the actives as well as the comments inserted by the users about the component in question. After submission, a reference for the active is inserted in the corresponding page with an observation indicating that it has not been approved yet.

Each active presents a set of characteristics that are grouped in a XML document. This document will be the basis for the routine of research and actives recuperation of ProReuse.

For the research the user may specify in which fields of the XML document he wishes it to be done. Thus, the user may eliminate unnecessary information and more easily locate the desired actives.

ProReuse may be accessed by public users (inside the intranet) or by users registered through the provision of login and password. Hence, the system offers a security control that works as follows: Only the registered users may submit components and documents, visualize and recuperate them. The public users may only perform researches, but they cannot recuperate the actives found. They are only allowed to access informative documents, such as readme, white papers and user manuals.

Still, the repository provides a catalog with all the available components. This catalog brings their description and a reference to their pages for the registered users.

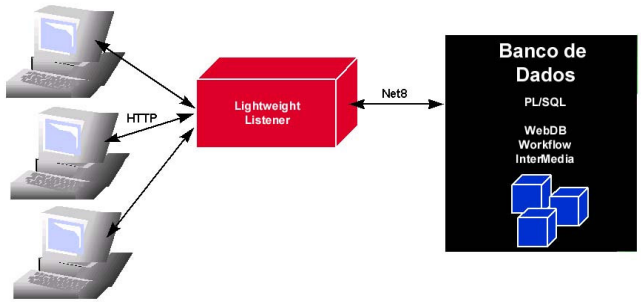
Finally, through ProReuse, it is possible to share experience through the insertion of comments. A user that has already used a component may insert relevant data on his experience and with the product and these data will be available for all the users so they can follow his recommendations.

The main functionalities available to ProReuse users are:

- Submission of Components and Documents – responsible for initiating the process of actives approval. In submission, the user provides the component's or the document's main characteristics that will compose the XML document.
- Processes Follow up – enables a user to graphically follow up all the processes initiated by him. The user may, at any moment, identify in which point of the process is the submitted active.
- Research – allows the search for the components and documents that compose the repository. This research may be contextualized, that is, it is allowed to the user to identify in which fields of the XML document he wishes to have the research done.
- Catalog Exploration – shows to the user all the components that are a part of the repository together with its descriptions.
- Comments insertion – allows a person who has already used a component or a document to insert comments on that active in a way as to share experiences.

3.3 Architecture

For the construction of ProReuse an architecture that is based mainly in the structure of a relational database was adopted. It integrates three tolls to this structure, according to Picture 1 – ProR.



Picture 1 – ProReuse Architecture

As we can observe, ProReuse corresponds to an instance of the bank that is accessed by client stations through a web interface listener. All its logic is stored in the bank in the form of PL/SQL proceedings that generate HTML pages to be visualized through a navigator in the client stations. The access to these proceedings is done through a web server represented by a HTTP listener grouped to a gateway PL/ SQL.

The elements present in this architecture are:

- Oracle WebDB – Tool used for the development of applications for the Internet. Provides to the developers functions as the applications monitoring and access control. It is stored in the database and PL/SQL packets represent all of its functionalities.
- Oracle Workflow - Toll used for the definition and automation of processes. Just like the WebDB, it is completely stored in the bank. It is composed of a managing mechanism that controls the execution of the process and by a notification system responsible for the sending of messages and for the processing of their answers.
- Oracle Intermedia – Tool used for the research of documents stored in the bank.
- Listener and PL/SQL gateway – These elements are a part of Web DB and act as the intermediary layer between the HTTP protocol used by the navigator and the NET8 protocol used by the bank. This allows the site links to be PL/SQL proceedings requisitions.

3.4 Processes

In this section, the workflow routine modeled for a components[4] approval process will be explained. The component submitted to the repository will pass through this process, which corresponds to the analysis of specialized managers that are a part of a hierarchy and of a quality group that evaluates the correctness of a component in relation to predefined patterns.

3.4.1 Process Components Approval

This process presents all the steps defined for the approval of a component in a higher level (see Picture). The process begins when the requester submits a component in the repository. The process' steps are:

1. Check if the requester corresponds to the last level of hierarchy (f[5]: Checks Level).
2. If it is the last, the flow will follow directly to the sub process of quality group analysis (p: Checks Quality- follows step 11).

[4] In the context of this process, the word component will be representing both a component and a document submitted to the repository.

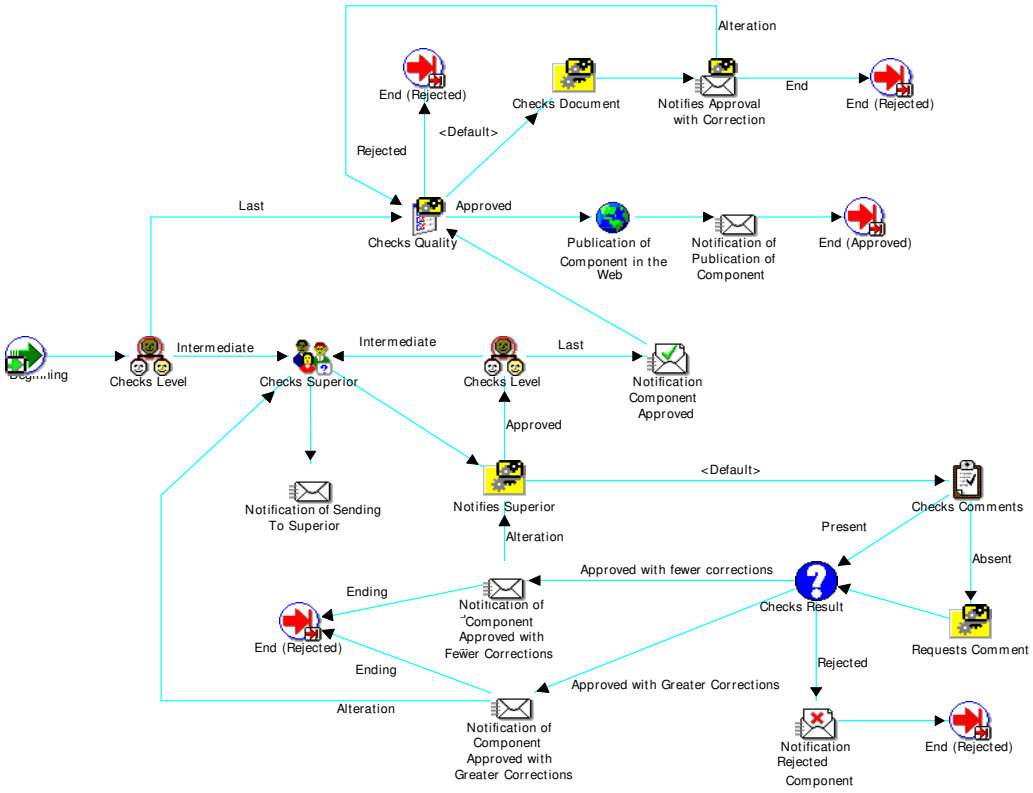
[5] Represents a function defined in the process.

3. If it is in intermediate level, it will be checked who is the immediate superior in the hierarchy (f: Checks Superior).
4. The notification sub process of this superior about the component submission requiring his approval (p[6]: Notifies Superior) is initiated. The requester is notified that his request has been sent for this superior.
5. The return of the previous process is expected. It corresponds to the superior's answer, and he can approve, approve with greater or fewer corrections or reject the component.
6. In case the request is approved, the level of the evaluator superior in the hierarchy will be checked (f: Checks Level), and if it corresponds to the last level, then the requester will be notified about the approval and the quality group notification sub process on the need to evaluate the submitted component will begin (p: Checks Quality – follows step 11). If the superior is in an intermediate level in the hierarchy, the immediate superior in the hierarchy will be checked and, following that, begins the sub process of notification of this superior about the component submission, which requires his approval (p: Notifies Superior).
7. In the cases where the request is approve with corrections or rejected, the insertion of comments will be required from the superior evaluator in order to justify his evaluation result. The insertion of comments is checked and, in case there are none, the process that notifies the superior about the necessity to insert comments is initiated (p: Requests Comments). When the superior inserts the comments or if he has already done it previously, the result is checked (f: Checks Result) and the requester is notified about the approval result.
8. In case the request is approved with greater corrections, the requester will have the option of altering the component and resubmitting it to his immediate superior. The requester will also have the option of terminating the process in case he does not wish to continue. If he chooses to do so, the process is terminated.
9. In case the request is approved with fewer corrections, the requester will have the option of altering the component and resubmit it to the superior that has analyzed it. The requester will also have the option of terminating the process in case he does not wish to continue. If he chooses to do so, the process is terminated.
10. In case the request is rejected, the requester is notified and the process, finished.
11. In the case of approval by the last superior in the hierarchy, the sub process that requires the quality group to evaluate the component will begin (p: Checks Quality). The group may approve, approve with corrections, or reject the component.
12. In case the component is approved by the quality group, the component is published in the repository site (f: Publication of Component in the Web), the requester is notified about the publication and the process, finished.
13. In case the component is rejected, the requester is notified and the process, finished.
14. In case the component is approved with corrections, the analyzer will be asked to inform the necessary corrections. The performance of this task is checked and, if it has not been done, its performance is demanded (p: Checks Document). The process will only continue when the corrections are inserted. When this happens, the requester will be notified about the result and he will have the option to correct the component or give up the request, ending the process. In case he chooses to continue, after correcting the component and

[6] Represents a subprocess of the main process.

resubmitting it, the process of quality group notification, asking for the component evaluation is reinitiated (p: Checks Quality).

Notifications for the requester, the superiors and quality group analysts are done via e-mail. In those that demand an answer, the message itself has an area reserved for its inclusion.



Picture 2 – Components Approval Process

Process Notifies Superior

Sub process of components approval that corresponds to the notification of a superior about the submission of a component that requires its approval. The steps corresponding to this sub process are described below:

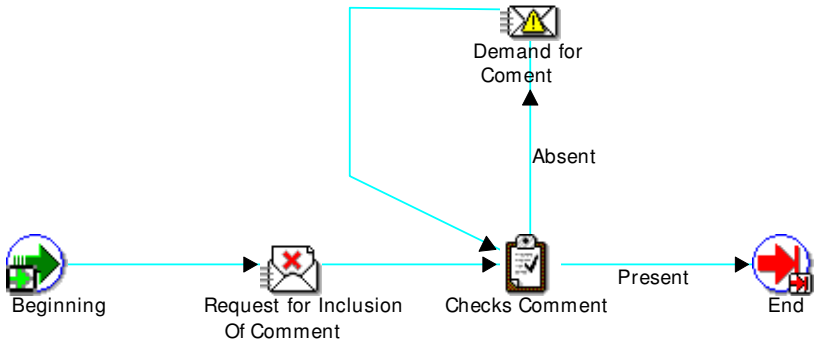
1. The superior receives the notification with submission data: component's name and type, requester's name and link for the component file. In case it is the submission of a document, the name of the component to which this document is associated is also provided.
2. In the notification itself, the superior answers to the approval request with the result of his evaluation. The result may be: approved, approved with greater corrections, approved with fewer corrections, and rejected.
3. When the superior answers it, the process is finished.

3.4.2 Process Requests Comment

Sub process of components approval that corresponds to the request of comments insertion by the superior, when he rejects or approves with corrections a component approval request (see

Picture 1). It was checked before that he had not provided any comments (f: Check Comment). The steps to the process are:

1. The superior receives the notification, remembering him the result given by him for a certain component and the necessity of providing comment, explaining the results given.
2. The superior will answer to the notification inserting comment in the space reserved for it.
3. Comment provision is checked (f: Checks Comment).
4. In case he has not provided it, a new notification will be sent and it will keep waiting to be answered.
5. When the insertion has really been done, the process will be finished.

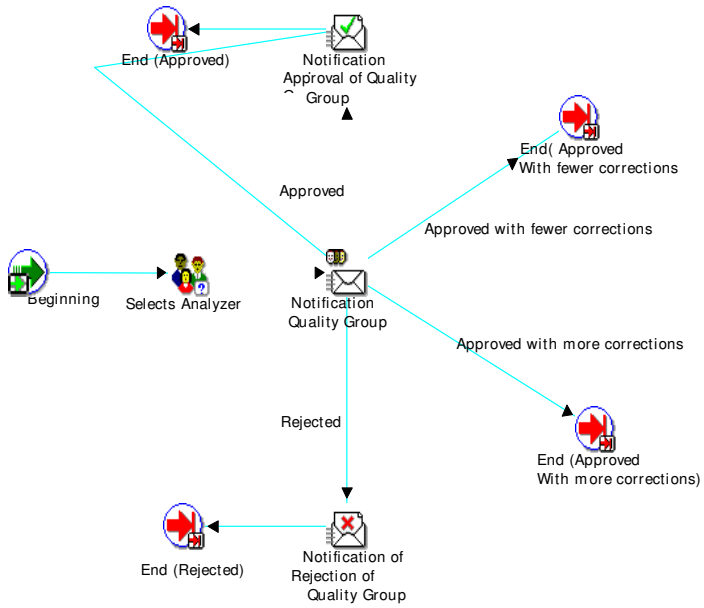


Picture 1 – Process Requests Comment

3.4.3 Process Checks Quality

Components approval sub process that corresponds to the component quality analysis process, after it has been approved by the superior (s) (see Picture 2).

1. Checks which is the analyzer correspondent to the type of component being submitted, since a certain member of the quality group is responsible for the analysis for each type of component (f: Selects Analyzer).
2. This analyzer receives the notification, requesting the quality analysis for the component and alerting about the need to insert a corrections document in case it is approved with corrections.
3. In the notification the component's name and type, the requester, and the link for the component's file are given. In case the submission of a document has been done, the name of the component to which the document is associated is also given.
4. In the notification, the analyzer will still be remembered that it is needed to insert a document with the corrections to be done for the approved with corrections type of result. In the notification message, there is a link to the form where this document may be submitted.
5. In the notification itself, the analyzer answers to the request with the result of his analysis. The result may be: approved, approved with greater corrections, approved with fewer corrections and rejected.
6. When the analyzer answers it, the process is finished.



Picture 2 – Process Checks Quality

3.4.4 Process Checks Document

Components approval sub process that corresponds to the process of checking and demanding a correction document from the quality group analyzer when he approves a component with corrections.

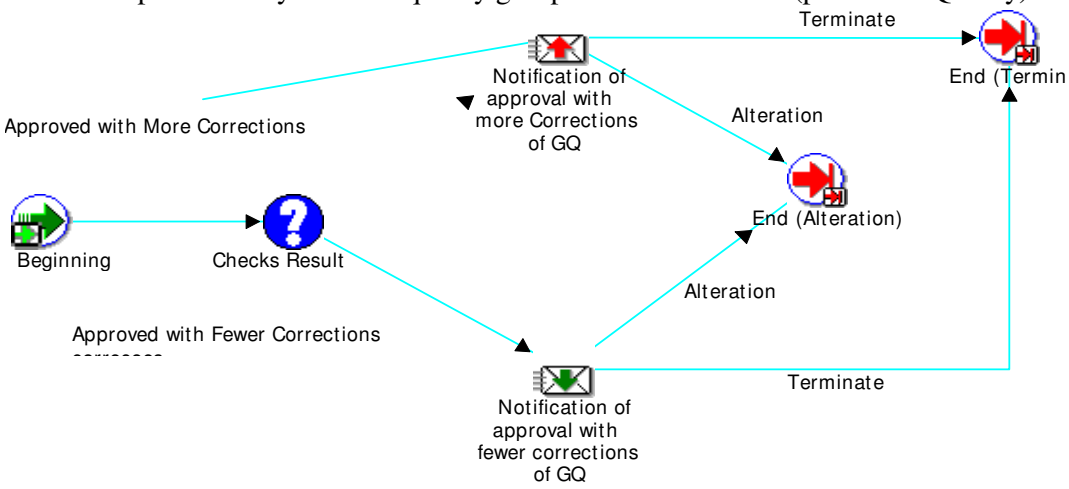
1) It is checked if the corrections document was inserted.
2) If it was inserted, the process is finished.
3) In case it has not been inserted, the analyzer receives the notification, remembering him of the need to insert a correction document when he approves with corrections and requests him to do so, due to the result provided by him for the component in question.
4) In the notification, the name and type of component are provided, as well as the requester, the link to the component's file and the link to the corrections document insertion form. In the case of document submission, the name of the component to which it is associated will also be provided.
5) In the notification itself, the analyzer answers to the notification after inserting the document through the indicated form, informing that the document has already been inserted.
6) When the analyzer answers it, the effective insertion of the document is checked to then return to step 1 of the process.

3.4.5 Process Notifies Approval with Correction

Components approval sub process that corresponds to the checking process with which type of corrections, greater or fewer, the component was approved by the quality group analyzer and the requester's notification about this result (see Picture 3). The steps of this sub process are described as follows:

1. The type of result given to the component, approval with greater or fewer corrections, is checked.

2. The requester receives notification informing the type of result obtained by the quality group analysis. In the message, the link containing the document with corrections submitted by the analyzer will be provided.
3. In any of the cases, with greater or fewer corrections, the requester will be able to choose between the option of altering the component and resubmitting it to the quality group or quitting the submission, terminating the process. In case he chooses to alter it, after answering the notification, the process of requesting a component analysis to the quality group will be reinitiated (p: Checks Quality).



Picture 3 - Process Notifies Approval with Correction

4 Conclusions

The increasingly competitive market in which software developing enterprises are placed nowadays demands a quite different behavior from the one practiced for many years. Their clients do not accept large and complex pieces of software that present many defects and do not fulfill the needs of users anymore. Due to these demands, the enterprises concerned about surviving in the market try to reach objectives that demand a lot of effort and require time to be achieved, such as: increase in quality, client's satisfaction, technical improvement, reduction of delivery time, increase in productivity and reduction of defects and risks. Moreover, with the increasing advance of technology, these enterprises need to adapt themselves to better attend all clients' expectancies.

To attend all these needs, reuse proposes the development of systems from preexisting software products, so that similarities between requisites and architectures of different systems may be better used. With the identification of similarities it is possible to reduce the time of delivery and the costs of development and also to increase the productivity and the quality of the final product.

In order to increase the practice of reuse in the organizations, many enterprises have used a repository where it is possible to retrieve pre-constructed actives that might simplify the development process and help in the dissemination of experiences and common practices in the corporative environment. Thus, a repository is a technical factor that introduces a reuse strategy adopted by an organization. However, other factors such as managing, motivation and training to use the repository are also vital to enable the use of this business strategy.

The implementation of ProReuse took into consideration factors that might be crucial to the success in the reuse process. Among them, we highlight: the analysis process; validation and

approval of components that make possible to guarantee the quality of what is being inserted in the repository; the contextualized research that allows a more accurate recuperation of the actives and its availability in web environment. This last factor allows the remote access to all the enterprise's employees.

Having in mind the enterprise's initial intention with the construction of an executable components repository, ProReuse was focused in the storage of these components and of the documents related to each one of them. Nevertheless, we have observed that it is possible and interesting to amplify it to a general knowledge repository where it its possible to insert not only software actives, but also proceedings inherent to a certain process, organization rules for the execution of activities, training materials and everything else that represents the informative patrimony of an enterprise. This is what the Knowledge Managing researchers propose (*Knowledge Management*). Another issue to be considered as a future adaptation is to allow the authors of the actives themselves to change their inventions according to the constant changes presented by the technological area.

Finally, we emphasize that the implementation of ProReuse has shown how the construction of a repository by those organizations that wish to achieve the benefits obtained with reuse is viable. This will enable small enterprises to also participate in the great enterprises group that has already benefited from this practice and to walk, together with Software Engineering, towards the excellence in system's development.

Bibliographic Reference:

[BADCOCK 1998] BADCOCK, Charles. Repositories Reap E-Commerce Role. Available on the Internet. http://www.zdnet.com/flitters/printfriendly/0,6061,2149796-35,00.html, 1998

[BASILI et al., 1996] BASILI, Victor R., Briand, L., Melo, W. L.. How Reuse Influences Productivity in Object-Oriented Systems. In: Communication of the ACM Number 10, October 1996. v. 39.

[BOOCH et. al., 1999] BOOCH, Grady, et al. The Unified Modeling Language User Guide. 1st.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.

[JACOBSON et al 1997] JACOBSON , I; Griss, ML; Jonson, P. *Software Reuse: Architecture, Process and Organization for Business Success*. Addison-Wesley-Longman, 1997

[KRUEGER 1992] KRUEGER, Charles W. Software reuse. In: ACM Computing Surveys. Number 2, June 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 Number 2, April 1997. v. 6, p.111-140.

[POULIN 1998] POULIN, Jeffrey S.. The Foundation of Reuse. Available on the Internet. http://www.umcs.maine.edu/~ftp/wisr/wisr9/final-papers/PoulinJ.html, 1998

[UREP 2000] UNISYS. Unisys Universal Repository. http://www.unisys.com/marketplace/urep, 2000