

Reengineering of Distributed Collaborative Applications

Bogdan VINTILĂ
 University of Economics, Romania
 Gothenburg University, Sweden
 vintila@chalmers.se

Dragos PALAGHITĂ
 University of Economics, Romania
 Gothenburg University, Sweden
 drapal@chalmers.se

Abstract: *The paper presents the collaborative environment. The particularities of the applications used within this environment are presented. The special requirements of the distributed collaborative applications are analyzed. A detailed description of the reengineering process is made. The workflow within the process is being followed from the start to the end. The causes that lead to the reengineering process are identified and measures of counterattack are built. The issues that must be considered in the reengineering process of the distributed collaborative applications are described. The main effects of the reengineering process of the collaborative applications are discussed. The costs of the reengineering process are estimated through a model of estimation. Future directions for the reengineering process of distributed collaborative applications are given.*

Keywords: *collaborative, reengineering, distributed, applications, requirements, model, team.*

1. Distributed collaborative applications

The collaborative term is one that appeared after the massive development of the communications technologies. Collaboration between human beings assumes the interchange of information. When a team is assigned a project, the work that will be done is collaborative, but the team members are not spread so they can communicate easily. When the team members are distributed within a huge geographical area, the communication issue tends to be more complex. Tools that automate communication processes have been developed after the wide spread of computer networks and these make the work of team members easier. A collaborative system is formed by the common goal, common resources, communication tools and the team members.



Fig. 1. Collaborative system

Fig. 1 shows the components of a collaborative system. The common goal is one thing that can't lack in the system. If the common goal is lacking we can't talk any more of a

collaborative system as each actor acts as an independent entity pursuing its own goals. The common goal is defined as the output of the project. It might take the form of a product, a service, a design, a process. The output of the project must fulfill the customer's specifications and expectations.

Common resources are allocated for the whole project and all team members must share them. In a project there are many types of resources. The human resources of a reengineering process in a collaborative system consist of the team members and project managers. The human resources are not to be shared between team members as they can't directly use other team members to do their activities. The resources that can be shared are materials, prime matter, equipments. These form a common resource pool that is accessed by all actors within the project. The resource manager is the one responsible for giving access to resources on the bases of the needs [1]. The resource pool must be used as efficient as possible to shorten the time everyone gets the resources they need and start producing. If the same resource is needed by more than one actor at the same time, the activities dependencies are taken into account when giving access to resources so that the workflow is optimized in report with execution time. If the dependencies of resources are not taken into account, an interlock of resources can occur as actors lock them in order to fulfill the goal of the activity, but their activity depends of an activity that needs the same resource and can't be finalized as the resource has been locked. The resource manager must be extra-cautious when assigning resources so that such interlocks don't occur.

Communication tools are very important within the team as these shorten the time needed for the information to be distributed among all team members. These tools are used for the exchange of messages, files and data. E-mail is one of the most known and used mean of transmission of information. Instant messaging is another service widely used by individuals to communicate short messages and solve small issues. Telephony is another comfortable mean of communication but the costs are usually higher than for e-mail or instant messaging. Telephony has the advantage of reaching the person more quickly than the other two. Electronic calendars are also good means of communicating deadlines and work in progress. Common calendars for all team members allow synchronization and short times of idleness.

Team members are the actors that share resources, communicate and complete activities in order to reach the common goal. These actors are assigned activities by the resource manager on the bases of their skills and competences [2]. The communication between team members is vital for the ongoing of the project they are all implied in. The lack of the communication between the team members manifests as delays in activities deadlines and dead times waiting for resources to unlock.

The informatics applications used in a collaborative environment have some particularities that are needed in order to fulfill the needs of the users. Regular applications can be used in such an environment after they have been modified to support the following:

- Save and e-mail function that ensures an increase in efficiency as the user no longer has to do two different activities but only one [3]; this also ensures that the user doesn't forget to e-mail the results to the interested persons;
- Communication means with the team members such as the ability to send messages, data files, reminders; this is vital in a collaborative environment as team members must change information very often to ensure a smooth ongoing of the project they are implied in [4];
- Centralized storage of files so that any user accessing a file has the last version of it; the centralized storage area ensures versioning of files and also the last versions for all users but also has the disadvantage of being vulnerable to attacks and

- malfunctioning; in order to solve this thread, the centralized storage area must be backed-up;
- Document management features are a must as every piece of information within a company must be accompanied by additional documents; the documents and information must be sent to different persons within the company;
- Meta data addition to normal documents and data files ensures the possibility to manage easily the documents and files using different criteria; finding data in the documents is done more easily as only some domains of data defined by a certain criterion must be searched within; also the meta data stores information about the persons that manipulated the document or file in time;
- Result reporting features are important for the evaluation of the project's progress; progress reports are important both in terms of total and partial achievements; the project's progress can be assessed by computing an aggregated indicator based on the partial progress of activities and the estimated time that has been allocated for each of them;
- Planning facilities that help the users stay focused on their work and activities; by planning the work and sticking to the plan, the team member fulfills deadlines and the outcomes have a high quality; sticking to the plan also gets the user accustomed to a certain work style and rhythm; this leads to the possibility of estimating better time needed for the future projects the team is assigned.

The applications that are used in the collaborative environment have some special needs. These needs are given by the multitude of requirements the users have. The requirements aim mostly the infrastructure but not only:

- Computer networks are needed for the transmission of messages and files; the save and e-mail function can't be used unless the computer the user works on has access to a network and a mail server;
- Communication is impossible if the network is not functional or there is no central server; a central server of communication is required within the company to ensure safe and cheap communication between the members of the teams; the voice-over-IP service is supplied for free by some servers; this is a way of ensuring very cheap voice communication within the company; a mail server is also needed as the files sent over e-mail are internal documents and must not be the subject of attacks;
- A central file storage server is needed to fulfill the space needs of users; this server must also have large bandwidth in order to be able to satisfy all the users in pleasing times; a second server for the back-up function is also required;
- A document management dedicated system is a must for large companies with hundreds of thousands of documents each year; the searching abilities of the server must be powerful enough to satisfy all requests; meta-data is a good way of narrowing the searching area;
- A database server for the authentication of users is also required; all resources are assigned and unlocked for users only after their authentication and authorization; in the lack of an authentication mechanism there are no measures of limiting file access and resource allocation.



Fig. 2. Requirements of collaborative applications

Fig. 2 gathers all the requirements of the collaborative applications in a graphical form. All these are important for the functioning of the applications and the lack of any of them leads to the malfunctioning of the collaborative environment.

2. The reengineering process

The reengineering process is necessary when the application no longer fits the needs of the users as requirements aroused in time. Informatics applications can't evolve and also can't determine the user's requirements so after some time they were developed these must be updated. The maintenance process assumes applying updates to the application so that it fulfills the new requirements of the users and the environmental changes. The maintenance process involves very low costs and also quick processes. When these processes prove to be costly and time consuming, it is the right time for the reengineering of the application. The reengineering process assumes:

- Identifying the problems the application is facing and their causes [5]; in order to solve a problem it is vital to have it identified;
- Finding possible solutions for the identified problems; the reengineering team must come with many solutions to the identified problems; the team members must be in touch with new technologies to be able to find usage of these and improve the applications they reengineer;
- Solutions filtering is necessary as not all are viable and also not all are possible at all; the solutions that imply the lowest costs and the shortest time for implementing are the ones preferred by the reengineering team;
- Applying the selected solutions to the application proves to be the most time consuming step of the process; at this stage the whole team mobilizes to modify the application in order to improve the performance and features;

- Testing the application after the reengineering process to verify that all features function as they should be is very important as the aim of the project is to improve performance and features, not to worsen them;
- Installing the application at the customer is the final step of the process; the customer's satisfaction with the reengineered application is an indicator of the quality of the reengineering process.

Fig. 3 shows the steps of the reengineering process and the order they are applied until



Fig. 3. Reengineering process

finalizing the process.

There are many causes that lead to the reengineering of the application. The continuous change of the environment and the increase in user requirements are causing many applications to evolve and develop more features. The main causes leading to the process of reengineering are:

- Environmental changes cause the most cases of reengineering through the appearance of new processing branches in the already existent applications; there are also many cases in which the application must be changed because of the changes of legislation;
- Quality requirements increase and the application must fulfill them in order to keep its customers; as users experience more and more applications, a collection of features and quality characteristics that must not lack any application forms; in time the collection increases and the users find the applications that don't have these elements to be displeasing;
- Technology advancement is another factor that leads to the reengineering process; the new technologies have the tendency of consuming less and less resources and performing better and better; the reengineering is needed in this case in order to reduce the functioning costs of the current applications and to increase the performance of the processing [6];
- High maintenance costs are faced by long-run applications; the applications that are updated for a long time grow in complexity and the maintenance process needs more time and resources; when the maintenance costs are too high the decision of reengineering the application is grounded;
- The lack of features the users require from applications that treat the same problem causes also the reengineering of the application [7]; if there are few features that lack, the process of updating the application is preferred as it is cheaper and less time consuming; if there are many features that need implementation and many changes are necessary to the application, the reengineering process is a good alternative as over time proves to be more cost efficient and less time consuming.

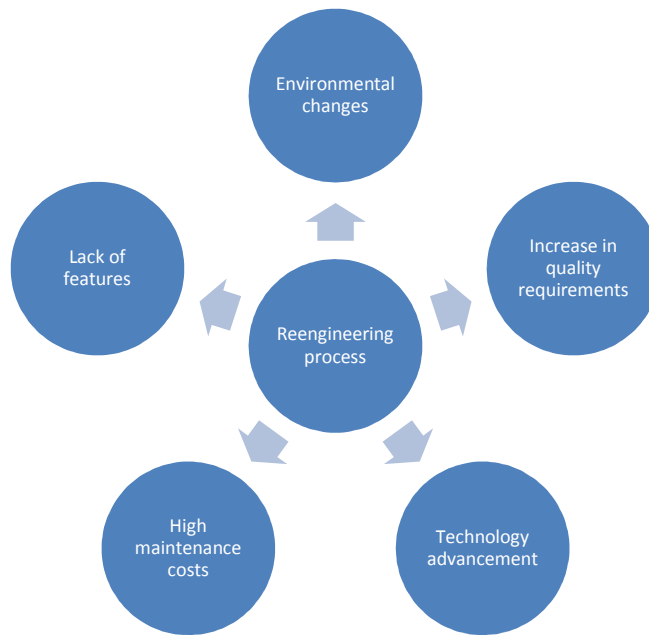


Fig. 4. Common causes of the reengineering process

Fig. 4 shows the most common issues the process of reengineering is caused by. In order to overstep these causes, measures of counterattack must be developed. Measures of improving the above issues are:

- A modular structure of the application makes the evolution much smoother and the environmental changes can be easily implemented; building modules separated by logic borders of processing is a good way of making things better; the maintenance process is eased of the burden of identifying the right code to change and the costs are kept low for a long period of time prolonging the life of the application;
- Quality requirements can only be fulfilled through the use of better components and ensuring a very detailed testing process; even if the used components are of high quality, the assembling phase can generate errors; through a detailed testing process these errors are identified and corrected ensuring a high quality for the final product;
- Technology changes can be taken by the application step by step; a modular structure allows the maintenance team to update one module at a time ensuring that the modules that benefit the most from the technological advancement are the first to be updated; the modules of the applications must also be grouped by the main technologies they implement so that when a new technology emerges, the maintenance team knows exactly what modules to update;
- High maintenance costs are not a problem for an modular application built using the object oriented model; by using inheritance there are many changes that are done in only one place and then the effects are seen in the whole application; code reutilization is another way of keeping the costs low; the use of open-source components in the applications gives the guarantee of having an in depth tested module and also the advantage of very low cost;
- The lack of features can only be counterattacked through a structure of the application that allows the addition of plug-ins and add-ons; using such an approach there will be very easy to add new functionality without high costs and with short implementing time.

The reengineering process is very complex as the structure of the applications is very diverse. Even though there are general principles to guide the developer in the reengineering fog, there is quite a difficult task to redesign the application and at the same time do it with limited resources, in a certain time and with an increase in quality and performance. The domain is as challenging as it is rewarding. The more complex the target application is, the more motivated the developer is to manage to improve it and add great features.

3. Particularities of reengineering collaborative applications

As seen before the distributed collaborative applications have special features and also require additional equipments to function at normal parameters. The reengineering process must take these special issues into consideration. Also, the aim of these application must be taken into account at the time of reengineering so that, after the process the applications are still oriented towards the collaborative work-style.

The reengineering process must consider the following when finding solutions for the problems the applications face:

- Distributed applications must serve a large number of users so the slightest improvement in performance can lead to a high economy of resources; the large number of users is also a disaster factor when the application has performance issues or memory leaks [8]; the malfunction of the application has many effect on a large number of users;
- Combining functions that are used together leads to greater user efficiency and also to a lighter interface; save and e-mail functions or scan and archive are examples of processing functions that are combined more than once;
- Major changes are not recommended as there are many users that must get accustomed with them; people don't like major changes and are likely to reject them; small appealing changes that serve the main goal and lead to the final result, if made over a long enough period of time, are more likely to be embraced by users; permanent change might also prove to be a problem as users can't rely on the application;
- As the applications are distributed they function over a network; the traffic of the applications is a very important factor of performance; the reengineering process must keep in mind that bandwidth is a valuable resource that influences the overall performance of the application and its users; the aim of minimizing the bandwidth consumption can be achieved through the optimization of the data flow, removal of unnecessary data, compression of digital content;
- Security is a very important issue as these applications are mostly used within companies and internal data must not be accessed by external entities without authorization; also, in the company there are many access levels that form a rule system for the access at the system's resources; security modules must be updated as frequently as possible to be able to face the newest threads;
- Collaborative tools that are newer and more efficient than the ones already existent bring increases in efficiency at the user's level; the implementation of many collaborative tools proves to be an easy process as they can be found as open-source components that integrate within the application; there are also many company specific items that can be created to improve efficiency; talking directly to the users of the current application and hearing their opinions about how the application can be improved can lead to some impressive results;

- Communication with other applications is very important for collaborative applications; distributed applications can take advantage of web services and implement communication services with other applications; these can be used by in-company applications but also by out-company ones; having a wide range of services that offer many features makes automated transactions work like a charm; these services must be secured so that only authorized applications can use them; in the case of insecure services, bad intended people can design applications that make many requests to the company's server in order to lock resources and finally shut down the whole system; the secure web service must identify the one making the request and grant access only if the credentials match; even in the case of a massive attack using requests to the server, the authentication service is fast enough not to slowdown the system and also there are no resources that are locked by the operation;
- Progress monitoring and control tools are to be implemented if they don't exist as these are of great importance for the upper management; if one knows the work progress of a project he can evaluate the efficiency of the team and also can program future project based on an simple prevision; the continuous control and monitor processes also lead to performance teams as by changing members the results are seen instantly and through experimentation and trials, the manager can build high performance teams.

Should the reengineering process take into account these issues, the resulting application is improved and its use in the collaborative environment leads to higher performance for all users. If these issues are not taken into account the reengineering proves to be a failure and the resulting application doesn't fulfill the user's needs.

4. Effects of the reengineering process

The reengineering process is considered successfully if succeeds to improve the application and solve the problematic issues. The effects of the reengineering process of the distributed collaborative applications are:

- Performance increases of the applications that lead to a lower resource consumption [9]; resources are very important and all companies struggle to consume less and quire more, thus the reengineering process creates resources through economy;
- More features are added to the application so the users can do more in less time with only one application; the need of different applications is lowered and also the time the user spends to use many applications and transfer results from one to another decreases leading to increased user efficiency;
- Future maintenance costs are lowered as the application gets again a logical structure; the logical structure of the application is altered by the repeated updates; these break the logical structure through additions of code; after many updates the source code is very complex and often doesn't follow any logical paths; the reengineering process solves the problem by reorganizing the code;
- New communication methods that emerge from the needs of the users; the collaborative environment must be improved through communication methods that the users can use in order to shorten the time needed for the transmission of information;

- A longer life cycle of the applications as once reengineered these function many years if maintained [8]; the life of the application gets a boost as it can satisfy users and function with low costs;

The problem of costs is very important in every domain. In reengineering is also very important to determine the estimative costs of the process. Cost estimation is a process that depends on many factors [10]. In the reengineering process the costs depend on how much of the application is affected and also by the complexity of the affected modules [2]. For the cost estimation there is also taken into account a medium cost of the medium complexity module of medium size determined by each company for its teams. The estimated cost is given by the formula:

$$Cost = \sum_{i=1}^n S_{m_i} * C_{m_i} * ra$$

where:

- n - the number of modules the application has;
- S_{m_i} - the size of module i ; this gets values reporting the size of the module with the medium set size;
- C_{m_i} - the complexity of module i ; this gets values reporting the module of the module with the medium set complexity;
- ra - ra has the value 1 if the module is affected by the reengineering process and has the value 0 if the module is not affected;

The estimated cost therefore will be higher if the considered application has many modules of high complexity that are affected by the reengineering process. The costs of the reengineering can be maintained low by designing modules as simple as possible. Lowering the number of modules affected by reengineering is also a good approach if it serves the final goal.

5. Conclusions

A collaborative system is formed by the common goal, common resources, communication tools and the team members. Common resources are allocated for the whole project and all team members must share them. Communication tools are very important within the team as these shorten the time needed for the information to be distributed among all team members. Team members are the actors that share resources, communicate and complete activities in order to reach the common goal. The informatics applications used in a collaborative environment have some particularities that are needed in order to fulfill the needs of the users. In order to be able to have these special issues the applications have also particular hardware and infrastructure requirements.

The reengineering process is necessary when the application no longer fits the needs of the users as requirements aroused in time. There are many causes that lead to the reengineering of the application. The continuous change of the environment and the increase in user requirements are causing many applications to evolve and develop more features. In order to overstep the causes of the reengineering process, measures of counterattack must be developed. The distributed collaborative applications have special features and also require additional equipments to function at normal parameters. The reengineering process must take these special issues into consideration.

The reengineering process is considered successfully if succeeds to improve the application and solve the problematic issues. The effects of the reengineering process of the

distributed collaborative applications are diverse and beneficial. The cost estimation of the reengineering process depends on the number of modules of the affected application and their complexity.

Further research aims at developing a methodology for the developing of collaborative applications projects. The development of a quality characteristics system that aims critical zones in the process of reengineering process of the distributed collaborative applications is to be done. The creation of a metrics system for the quantitative analysis of the progress from the reengineering process is aimed after the quality system is finalized.

Acknowledgements

This article is a result of the project „Doctoral Program and PhD Students in the education research and innovation triangle”. This project is co funded by European Social Fund through The Sectorial Operational Program for Human Resources Development 2007-2013, coordinated by The Bucharest Academy of Economic Studies.

References

- [1] A. Söderholm, "Project management of unexpected events ," *International Journal of Project Management*, vol. 26, no. 1, pp. 80-86, Jan. 2008.
- [2] J. Thomas and T. Mengel, "Preparing project managers to deal with complexity – Advanced project management education," *International Journal of Project Management*, vol. 26, no. 3, pp. 304-315, Apr. 2008.
- [3] C. G. a. S. Weibelzahl, "Usability Engineering for the Adaptive Web," in *The Adaptive Web*. Springer Berlin / Heidelberg, 2007, pp. 720-762.
- [4] U. M. Krause, R. Stark, and H. Mandl, "The effects of cooperative learning and feedback on e-learning in statistics ," *Learning and Instruction*, vol. 19, no. 2, pp. 158-170, Apr. 2009.
- [5] R. Koschke, "Software visualization in software maintenance, reverse engineering, and re-engineering: a research survey," *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 15, no. 2, pp. 87-109, Apr. 2003.
- [6] C. D. Rosso, "Software performance tuning of software product family architectures: Two case studies in the real-time embedded systems domain ," *Journal of Systems and Software*, vol. 81, no. 1, pp. 1-19, Jan. 2008.
- [7] M. C. Feathers, *Working Effectively with Legacy Code*. Prentice Hall, 2005.
- [8] S. Valenti, Ed., *Successful Software Reengineering*. IRM Press, 2002.
- [9] D. P. G. A. L. Robert C. Seacord, *Modernizing Legacy Systems: Software Technologies*,

Engineering Processes, and Business Practices. Addison-Wesley, 2003.

- [10] P. Pocatilu, *Costurile testarii software*. Bucharest, Romania: ASE Publishing House, 2004.

Authors



Bogdan VINTILĂ graduated the Bucharest University of Economics, the Faculty of Cybernetics, Statistics and Economic Informatics. He is currently a PhD candidate in the field of Economic Informatics at University of Economics and at the University of Gothenburg in the Applied IT department. He is interested in citizen oriented informatics applications, developing applications with large number of users and large data volumes, e-government, e-business, project management, applications' security and applications' quality characteristics.



Dragos PALAGHIȚĂ graduated from the Academy of Economic Studies of Bucharest, Cybernetics Statistics and Economic Informatics faculty, Economic Informatics section in 2008. He is programming in C++ and C# and his main areas of interest are Informatics Security and Software Quality Management. Currently he is undergoing PhD studies at the University of Economics and the University of Gothenburg within the Applied IT department. He published 14 articles in JAQM, Informatica Economică Journal, Economie Teoretică si Aplicată Journal, Revista Română de Automatică si Informatică. He is an editorial board member for the Open Source Science Journal.