A Mexican Experience Redesigning a Software Development Process Using XP, FDD and RUP

Luis Carlos Aceves Gutiérrez
Universidad de Monterrey
Av. Morones Prieto 4500 pte., 66220, San Pedro Garza Garcia, Nuevo León, México
http://www.udem.edu.mx/
laceves@udem.edu.mx

Enrique Sebastián Canseco Castro
Universidad de Monterrey
Av. Morones Prieto 4500 pte., 66220, San Pedro Garza Garcia, Nuevo León, México
http://www.udem.edu.mx/
ecanseco@yahoo.com

Mauricio Ruanova Hurtado
Universidad de Monterrey
Av. Morones Prieto 4500 pte., 66220, San Pedro Garza Garcia, Nuevo León, México
http://www.udem.edu.mx/
mruanova@yahoo.com

Abstract

This work focuses on the process redesign for software development and management implemented at a Mexican steel manufacturing company. The main goal was to improve the software systems production, maintenance and support; it was achieved by implementing artifacts and practices of RUP and Agile Methodologies like FDD and XP. Throughout this document we can identify the needs and problems on the software development and management faced by the steel manufacturer. To correct these, Business Process Reengineering was adopted, specifically Process Redesign (BPR). We also present some results on the implementation of BPR through its application on some manufacturer's information systems.

Keywords: Business Process Redesign(BPR), Rational Unified Process (RUP), Extreme Programming (XP), Feature Driven Development (FDD)

1. INTRODUCTION

Trying to change a large organization in Latin America is difficult. There is a report of Gartner Group that foretells the growth of information technology in Latin America. This growth is around 134 billion dollars [7]. As a result of that report, a lot of large Mexican companies have been trying to maximize the use of information technology to become more efficient.

The organization in question is a large steel manufacturing company. It exports worldwide and has a European partner. Information systems are important assets used to improve steel production and management. The company has an IT department with a software development and management area. There were no customary procedures in daily operation of the department and in most of the cases there was lack of formal documentation for their processes. As a result, software was developed with inadequate or nonexistent documentation.

This paper will briefly outline the analysis of that development and management of information systems. We will describe some problem indicators. We will then consider the process to choose and adapt some software development methodologies to improve the processes. We will show the business process redesign using the chosen methodologies, and finally we will conclude with some considerations and results on the implementation of the redesigned process versus the original process.

2. Background and Diagnosis

In order to establish the situation, interviews, surveys and formal and informal meetings were carried out.

The software construction was centralized. The main tools used were Centura Software for the front-end and SQL Server 7.0 for backend. [14] All applications were developed as client/server architecture. Some of the modules of these applications had as much as seven years in the production stage.

The company had a Document Manager System (SAD) for document control. It was developed in-house and used for the enterprise policies documents only. However, the IT Department didn't implement SAD, and the documentation of its projects was distributed throughout the personal computers of the IT personnel. [8][12][14]

Three main processes were identified: New Projects, Project Change and Improvement, and General Support. New projects: involves the development of a solution which does not include existing modules. Project changes and improvements: involves modifications that users request and non-structural changes to existent applications and queries. General support projects: involves record and follow-up of system errors notifications, and technical questions.

2.1 Problems related with New Projects. Requirements were gathered by two persons. One of them handled the administrative systems and the other one the operative systems. The list of requirements was prioritized before sending it to the IT Department.

Periodic meetings were held by the requesting area and the software development personnel, in which a pre-analysis document was drawn. This document included, among other things, the problem description, the list of requirements, flow charts, database diagrams, site implementation and a list of people involved. A feasibility analysis was drawn, but no return of investment analysis was generated. [8][12][14]

Usually, functional prototypes were used as exploration, and helped define new requirements and functionality. Prototypes were developed in stages, and for each one a time and resources estimation was made. At the end of each stage, estimations for the next stage were adjusted. After several meetings, user approval was reached, and the system was implemented in all sites that needed. [8][12][14]

In the case of New Projects, three departments were working in a totally different manner, although the same results were generated.

2.2 Problems related with Project Change and Improvement. As in New projects process, requests were obtained through periodic meetings held with area representatives.

Sometimes modifications were grouped in blocks. All pending modifications of a system were addressed as a whole, thus isolated requirements where not processed. On other occasions daily time was allotted to improvement, and requirements were not grouped by system.

In all changes that affected a database, a form was filled out with information such as problem description, list of requirements, flow charts, database diagrams, site implementations and people involved. [8][12][14]

2.3 Problems related with General Support Projects. The company had a support area that included four persons. They were in charge of registering and following up system errors notifications and technical questions. [8][12][14]

An intranet Help Desk System was available, where the Support Department logged all user and application events for follow-up. Urgent issues were directly and immediately attended or could be addressed with the development team; issues on hold were registered on a to-do list to be attended. Each time a new version of any application was liberated, users and Support were notified. [8][12][14]

Indicator / Problem	Department is highly dependent on IT personnel	Difficult to adapt to organizational changes	Difficult systems maintenance	Development errors caused by IT department	Job posts without scope responsibilities	Non-documented or informally documented systems	Nonexistent formal development methodology.	Non-standardized development processes	Too much time invested studying system's code
Coworkers' collaboration								N C S	
Responsibilities scope					N C S		N C	N C S	
Task continuation					N C S			N C S	
Interface between personnel tasks		N C S					N C	N C S	
On time, truthful, and reliable documentation				N C		N C	N C	N C	
Input and output documentation							N C	N C	
Subordinate coordination							N C	N C	
Version control				N C		N C			
Centralized documentation				N C		N C	N C		
Employees training time	C		C S						C

Table 1. Summarization of problems and their indicators for each of the stages: New Projects (N), Change and Improvements (C), and General Support (S).

3. Choosing the Methodology to Implement

One of the main challenges of the project was to choose the software development methodology. To choose the best option, the following questions were answered:

What do we really want? : quick response changing customer demands and high productivity.

What do we already have? : close customer contact, concern for quality. There was software such as MS Project, ERWin, MS SQL Server and MS SharePoint.

What are our options? : RUP was chosen for being an object oriented development methodology. RUP facilitated one of the main goals of the organization: the migration of software development to C# technology. In case of organizational changes in the IT department, RUP allowed the continuous management of software development. Furthermore, RUP documentation could make feasible future certifications on CMM or other international standards [5].

FDD and XP practices, techniques and agile modeling artifacts were chosen because they supported codification by inspections, refactoring, cross reviewing, pair programming and programming standards, among others. [1][3][6][10]

RUP, FDD and XP practices, techniques and artifacts were redesigned internally to align them to the requirements of the software developed in the company. Some items were eliminated when they did not improve software development. Items were added to the practices, techniques and artifacts that proved to be useful to the software development process. None of these methodologies was completely adopted, they were adapted on a redesign of the software process development.

4. REDESIGNING THE SOFTWARE DEVELOPMENT PROCESS

Once the decision to use a mixed methodology based on XP, FDD and RUP was made, the next step was redesigning the software development and management processes including documentation, practices and work flows.

RUP artifacts were adapted to achieve formal software developing without generating a larger or more complex documentation. [10][11].

As we have seen, the documentation redesign identified the RUP, XP, and FDD artifacts and practices to implement. [1] Also, some of the original practices and documents internally developed as well as a set of useful software tools were kept. These were adapted and approved by consensus. Coding practices and bug diminution focused test were selected form XP and FDD. [4][9]

Finally, all artifacts, practices, tools and documents were cataloged according to RUP disciplines to standardize the development and management of software as figure 1 shows.

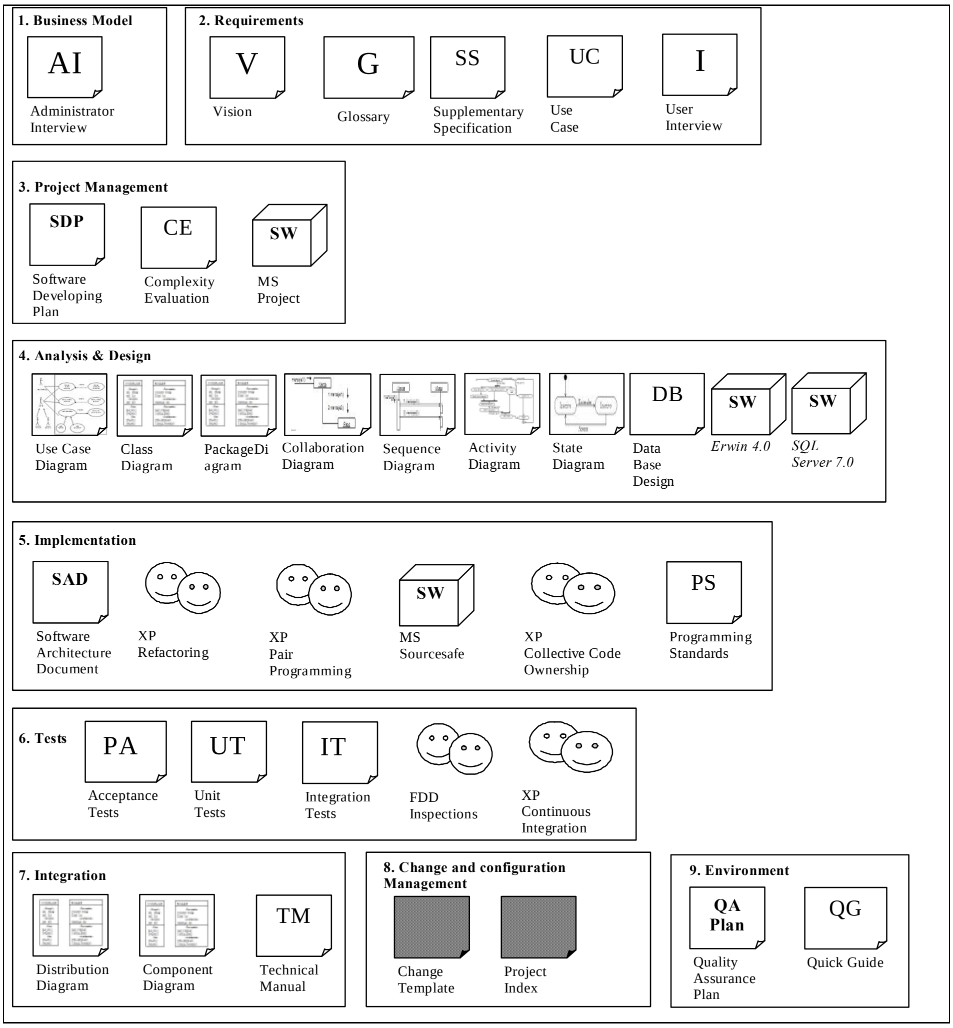

Figure 1. Summary of XP and FDD artifacts, documentation, practices and software tools implemented according to RUP disciplines.

5. RESULTS ON THE USE OF RUP, XP AND FDD

To carry out the model on figure 1, the redesigned processes were put into practice. The processes redesigned were implemented on projects that matched each category:

The following table shows the comparative time between the original vs. redesigned process. As we can see in New Projects the redesigned process was improved. However, in Change and Improvements the redesigned process took longer than the original one.

Activities	**New Projects**		**Change and Improvements**	
	Original	Redesigned	Original	Redesigned
Needs identification	36	10	6	8
Requirements description and deadlines	10	10	6	8
Analysis and design elaboration	40	40	3	15
Implementation	24	20	2	4
Total hours	110	80	17	35

Table 2. Comparative periods of development time between original and redesigned process in New Projects and Change and Improvement processes

In the case of New Projects the improvement on the performance can be explained by a bottleneck that occurred in the Vision document. It required constant revisions before liberation. This revision caused delays when meetings were not been programmed beforehand. There was no point in beginning a Use Case until the Vision document has been approved. This was the case of the User Interface Prototype and the database design, which were not begun until the Use Case was finished. In the database design the bottleneck originated problems when identifying the fields that the User Interface Prototype required. It became necessary to determine which fields already existed in the database. If a table modification was needed the delay was bigger and affected systems in production.

In the case of Change and Improvement the growth of time was because analysis and design were almost inexistent. Also there was a lack of documentation related with the change. With the redesigned process is mandatory to document the change using the artifacts and tools of the model on figure 1.

For the General Support process there was not improvement between redesigned vs. original process. This can be explained since people continued maintained their way of working, and just began filling new documentation.

Activities	**General Support**	
	Original	Redesigned
User service	2	2
System errors documentation	8	8
Total hours	10	10

Table 3. Comparative period of service between original and redesigned process in General Support process.

6. CONCLUSION

Although not all of the artifacts were included, this redesign did not speed up the development process. Nevertheless, it helped the company in the standardization of its development processes as well as in the tasks among the development team.

Redesigned allowed to differentiate clearly the employees responsibilities and functions, as well as whether their work was accomplishing what it had to. Thus, it became possible to take corrective actions whenever bottlenecks were reached.

Redesigned documents allowed to directly attack user needs, fulfill the task without working more than was needed and solving problems that were not user needs. They proved the project dealt with the user specifications.

Through process redesign it was possible to standardize work flows, actors and documentation within the IT department.

With the processes redesign it was possible to identify areas of opportunity within the process by elimination, simplification or integration of documents and responsibilities.

References

[1] Ambler S, “Agile Modeling, Agile Documentation”, http://www.agilemodeling.com/essays/agileDocumentation.htm , (2005)
[2] Action Technologies Inc. “Action Workflow Methodology” http://www.actiontech.com/about_us/methodology.html , (2003)
[3] Clipcode Knowledge Services, “Choosing a Software Development Methodology”, http://www.clipcode.biz , (2005)
[4] De Luca, J. y Coad, P., “The Community Portal for al things FDD”, Feature Driven Development, http://www.featuredrivendevelopment.com , (2003)
[5] Doyle K, "Achieving CMM Level 2 & 3 with RUP", http://www.ortia.com/cmm_rup_rug.pdf , (2003)
[6] Extreme Programming “Extreme Programming: A gentle introduction”, http://www.extremeprogramming.org , (2003)
[7] Gartner Group VIII Conferencia Anual, “Future of IT: la Justificación Económica de la TI”, http://www.pergaminovirtual.com.ar/revista/cgi-bin/hoy/archivos/2005/00000553.shtml ,(2005)
[8] González P, Personal interview, Project leader Deacero S.A. de C. V., Monterrey, N. L., 2003.
[9] Jeffries, R., “What is Extreme Programming?”, XProgramming.com An Extreme Programming Resource, ronjeffries@acm.org, http://xprogramming.com/xpmag/whatisXP.htm , (2001)
[10] Kruchten, P., The Rational Unified Process an Introduction, Addison-Wesley Pub Co., United States, (2000).
[11] Larman, C., Applying UML and Patterns, Pearson Education, Second Edition, India, (2002).
[12] Martínez, J. L. Personal interview, Systems Management, Deacero S.A. de C. V., Monterrey, N. L., 2003.
[13] Peppard, J., Rowland, P., La Escencia de la Reingeniería en los Procesos de Negocios, Prentice Hall, México, (1996).
[14] Ravizé, J. Personal interview, Systems Managements, Deacero S.A. de C. V., Monterrey, N. L., 2003