PROJECT MANAGEMENT MODEL FOR A PHYSICALLY DISTRIBUTED SOFTWARE DEVELOPMENT ENVIRONMENT

Roberto Zanoni
Scholarship Holder of DELL/PUCRS E-business Research Center
Master Student at PPGCC – Graduate Program in Computer Science
Pontifical Catolic University of Rio Grande do Sul, Porto Alegre, Brazil
puc_roberto_zanoni@dell.com

Jorge Luis Nicolas Audy
Professor at FACIN – Computer Science School, Ph.D.
Pontifical Catolic University of Rio Grande do Sul, Porto Alegre, Brazil
audy@inf.pucrs.br

Abstract

The objective of this study is to propose a project management model which includes the unified process (UP) and UML language, for e-business software development in a physically distributed environment. The research is a qualitative study, aiming to develop new methods and models. The main research method is the case study, according to Yin (1992). Initial results point towards a model, which includes the spiral life cycle type, the development process of object-oriented systems (using UML specification language and the unified process as proposed by RUP), and the incorporation of the procedural approach proposed by PMBOK. The proposed model is divided into 6 phases, where each one of these phases contains a set of associated activities. In the future, the intention is to develop a support software for the model and apply this software into the environment in study. This extremely important study line demonstrates that the business world and business practices in the software development area are moving ahead of existing theories and conceptual models in this area. This study, yet in its initial phases, starts to bring significant results that contribute to meet market expectations nowadays.

1. Introduction

Project Management is the application of knowledge, abilities and techniques to plan activities intending to meet necessities and exceed expectations of people involved, in relation to the project [PMB 00]. The act of meeting or exceeding necessities and expectations of people involved, always depend on the balance among concurrent demands: scope, deadline, cost and quality; different needs and expectations of people involved, and real needs and expectations.

According to [MER 00], project management derived from the demand of new management methods. The three basic strengths pushing the application of project management are: exponential growth of human knowledge; the emerging demand for more complex and standardized products and services; and the evolution of global competitiveness for developing services and products.

The software development project manager must bear in mind that its activity must pursue quality, productivity and risk reduction planning and executing the development of the product [PMB 00]. The software development is characterized, thru several studies, as a technical activity disassociated from the organizational sphere. Several efforts can be identified in the sense of defining, establishing and spreading information systems development approaches that ensure greater adherence to user’s specifications and necessities and alignment to the company’s business strategy. There are many system development approaches. Different problems and challenges have specific features that demand specific approaches. The major challenge is to select, adapt and integrate these approaches, according to the characteristics found in certain environments. The software development area requires the usage of a modern approach associated to project management, which meets the demands of a physically distributed environment. Among these approaches the object-oriented can be highlighted, and more specifically, the UML (Unified Modeling Language) and the UP (Unified Process).

The globalization of the economy has lead to several organizations to distribute geographically its resources and investments aiming to obtain better results. Physically distributed environment is a new tendency of software development, in which users and development teams are in different places, sometimes in different countries with diverse cultures. The software development area has been the first one in this distribution process, in countries such as India and Ireland. Greater productivity, cost and risk reduction and quality improvement, are some characteristics that large organizations pursue when spreading its software development processes offshore.

The e-business area stimulates even more the distribution of actors involved in the system process usage, definition and development. To organize and assure the quality of this integration, it is important to
have model with description capabilities and computational automation environments, for them, enabling the usage of these sort of descriptions for the computational workflow support.

The objective of this study is to propose a model that includes the unified process (UP) and UML language, based on PMBOK, for application in the area of e-business software development in a physically distributed environment. This study presents in section 2 a theoretical basis about project management. Section 3, presents the research method to be used and the research issue. The study case is described in section 4. Section 5 describes the proposed project management model. Section 6 brings the final considerations. Finally, section 7 presents the bibliography used throughout the research work.

2. Project Management

Several project management models have been analyzed ([CAN 98], [DUN 96], [FRA 94], [PMB 00]). After this first analysis, the decision was to deepen the studies of two of these models. The model 1 is the Procedural Project Management Model [PMB 00]. The model 2 is the Object-Oriented Project Management [CAN 98]. These models were chosen due to their characteristics (differences and similarities) that are considered adequate for the study being analyzed.

The model to be studied will be presented next.

2.1. Model 1: Procedural Project Management

Before starting planning a project, you must establish project’s objectives and scope, consider alternative solutions and identify technical and managerial limitations. Without this kind of information, it is almost impossible to define cost estimates, distribute project tasks or allocate efforts that provide a progressive evolution impression. The objectives setting establishes general targets to be reached in a project, without considering how these steps must be accomplished. This scope identification must be one of the first things to be appraised, specially if possible to conform it according to quantification functions. Once objectives and scope are set, alternative solutions can be considered, since they allow the designer to choose the best approaching, or at least a great approaching, due to deadlines restrictions, budget, labor force availability, technical limitations etc.

There are many classifications for the processes that compose project management. In [PMB 00], project management consists basically of initiation, planning, execution, monitoring and closing processes of the project. The initiation considers the necessary processes to assure that the project will be accomplished. The planning process aim to plan and keep a possible work scheme to reach the objectives of the project, involving scope setting, activities planning, the budget estimates and the project plans. The execution consists of coordinating people and resources to execute the plan and involves quality guarantee, information distribution and providers selection. Monitoring processes aim to assure that project objectives are being reached, through monitoring and appraisal of its progress, controlling changes, costs, quality and risks. Finally, the closing formalizes the project or phase acceptance, ending it organized.

These processes overcome the project life cycles, in a way that the closing of a phase enable the start of the following one. Besides, the three main processes: planning, execution and monitoring are recurrent, that is, in a certain phase of a project life cycle occur interactions among these processes until the phase can be finished. Figure 1, shows the connections between processes groups in each phase.

![Figure 1. Connections between processes groups in each phase](image)

PMBOK proposed by PMI (Project Management Institute), contemplates nine distinct areas of project management that describe the knowledge and practices of project management in terms of processes that are part of it. These areas are: project integration management, scope management, human resources management, time management, cost management, quality management, communication management, risks management and project acquisitions management.

2.2. Model 2: Object-Oriented Project Management

To manage software development teams it is necessary to be aware of how people communicate efficiently, because an inappropriate communication model set by the manager have been the reason of many project failures. Objects provide flexibility and control necessary to deal with requirements involved in a project management. The static and dynamic requirement descriptions provide an ambiguous description of the system, which both developers and clients are able to understand.
The UML (Unified Modeling Language) is a recent synthesis of object-oriented languages. It includes a set of consistent diagrams that can be used to describe and communicate system requirements of a software, project and code. It can be used to provide insights of the project vision and project requirements of different abstraction levels, and its artifacts provide a common view that works as basis for the collaborative project.

The UML focuses primarily on specifying and documenting system requirements and projects. It is important to keep using UML during the development process. If used in the initial phases only and do not connect UML artifacts to build and test activities, it is possible to lose control of the project.

The project manager must establish a structure for the precise communication among the team, and then insist with its usage. The implementation of a modeling language of standard object is a component of this structure. UML provides a number of static and dynamic diagrams that specify not ambiguously the system and its behavior. It deals about a precisely and clearly way to capture and share requirements. Thus UML, if used properly, provides documentation that eases the communication with clients as well as with team members.

Figure 2 illustrates the division and composition of the phases in the management process.

![Figure 2. Phase of the management process](image)

Due to client’s demands that always look something more, better and faster. We face the need of creating more complex software each time. The presence of highly trained developers is not enough. It is necessary an organizational guide – a process. It is necessary a process that integrates diverse development facets. UP (Unified Process) is one of the solutions presented for these problems.

UP is a generic framework of a development process, based on components and it uses the entire UML definition. UP is guided by use cases, centered in the architecture, interactive and incremental.

A project consists of phases, and the set of these phases is known as the project’s life cycle. The life cycle is used to define the beginning and the end of a project [PMB 00]. The software development literature has a variety of life cycle models: cascade, evolutionary, incremental, spiral. All life cycles have scope, draft, implementation, and verification activities, but they differ in how they schedule and organize the practice of these activities [DUN 96].

The life cycle of the unified process is divided into four phases, where each phase is subdivided into iterations. The first phase is the conception, where the project’s scope definition and the development of the business cases are done. The second phase is the elaboration, which involves the project plan, attributes specification and architecture sedimentation base line. The third phase is the construction, where the product is built. The fourth and last one is the transition, where the product is delivered to the user.

2.3. Comparative between the models presented

Having the proposed work environment (physically distributed) as reference, the intention was to detail the main contribution of both models studied. The following chart brings a comparative between these cited models, with important characteristics of both.

<table>
<thead>
<tr>
<th>Models</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project division into phases</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Quality oriented projects</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Spiral Life Cycle</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Object – oriented</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UML management</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Usage of objects</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Abstraction usage</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Embedment and package usage</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>United process adherence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial phase</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Elaboration phase</td>
<td>N</td>
<td>Y</td>
</tr>
</tbody>
</table>
None of the models studied presents support for a physically distributed software development environment, where the development team is distant from its users. Sometimes the distance is not physical only, but cultural as well. Due to this gap found in these models, it is suggestible to propose a project management model that supports a physically distributed environment and that uses object-orientation as basic development methodology (UML e UP).

The work process in the Research Center is set as follows: Austin and Porto Alegre teams develop the planning process together. Austin team identifies user’s needs and sends them to Porto Alegre as new systems development demands. Porto Alegre team does an initial analysis and assigns a Project Manager. Thus, the project is assembled and requirement specification process starts. The project team consists of a group of people who should work cooperatively, where the result of this work must be the product specified by the user.

The structure consists of technical and administrative coordinators, placed in Austin (USA) and Porto Alegre (BR), the project manager, architect, developer, database designer, quality analyst, technical writer and system administrator.

Following there is a description of two case studies observed during 2001.

4.1. Case Study 1: Call Center

In relation to the Call Center, when known of the necessity of creating a way to measure the kinds of problems related by costumers of certain types of products offered by a company, the utilization of an application to be used by a call center was suggested, that is, a center for service support. This application should allow phone call subscription, including user details, problem described and solution suggested by the attendant (in this case, employees in charge of solving technical difficulties related). Understanding the system to be implemented, we pass for its construction.

4.2. Case Study 2: E-Mail Blitz

Using as perspective the company’s business area, that is, e-commerce of computer products, advertising is necessary to be done through the WEB, to inform subscribers (costumers who have already bought something on the company’s website or have subscribed their interests on certain products) about these products releases. To do this, a new software system (E-Mail Blitz) was designed, where it would be possible to create an offer (composing messages about products), set a client or subscriber profile (that would be created according to their characteristics) and, finally, send the offer for a certain kind of profile via e-mail.

<table>
<thead>
<tr>
<th>Construction phase</th>
<th>Y</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transition phase</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Control Process</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Planning Process</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Accomplishment process</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Closed Process</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Project management view</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Support of physically distributed environment</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>
5. Proposed Model

Having the lived experience in the cases described and in the theory studied for perspective, a preliminary model that tries to answer the research issue presented in section 3 is presented.

In this sense, we’ve incorporated contributions of the project management approaches studied, according to the following presentation:

- Spiral type life cycle.
- Object-oriented system development process, using UML and UP specification languages.
- Incorporation of the procedural approaching proposed by PMBOK, expanding the management areas indicated.

The proposed model is divided into 6 phases; each one has its function and exit criterion. The first phase is the requirement definition. This phase starts after the development plan of software is written. Primarily a business requirements analysis is done, then system, subsystem and integration requirements are analyzed respectively during the phase. The exit criterion of this phase is the approval of the conceptual studies.

The second phase is the project exploration and definition. After approving conceptual studies, the phase is started. The first step of the conceptual project delineation, in which the team must be aware of the system understanding. Three other projects are involved in this phase: logical project, physical project and final projects. The exit criterion of this phase is the approval of the development.

The third phase consists of the production processes. After approving project development, the phase is started. The concept construction of a project (concept proof) is part of the beginning of this phase. The phase consists of: a primary production that is repassed to components test; a second production will be tested as well; and finally, the final production. The exit criterion of this phase is the approval of the production.

This fourth phase is the evaluation. After approving production, the phase is started. After concept construction is done, the risk analysis is performed. This analysis does the maximization of positive events results and the minimization of negative events consequences. Component and subsystem tests are also done in this phase after first and second production respectively and, the assessment and adherence is done after the conclusion of the project final production. The exit criterion of this phase is the approval of the main changes done, when needed.

The fifth phase is the transition, where the passage from the assessment and test phase to the release phase is done. Besides the system approval through tests, the client approval of the system set this phase objectives. The adjustment between the business process supported and the necessary procedures to make it operational are performed.

The sixth and last phase is the integration. This phase starts after tests are done and the main changes done are approved. The critical requirements to do this project is reliability, data integrity and project performance. The final model comes with the integration of the developed software and the business process to which it will be given support.

This model defines a set of procedures aiming to standardize and systemize the requirement specification in the interaction process between work teams physically distant from each other (Austin and Porto Alegre).

As defined before, the basis of this project is the PMBOK, incorporating the UML specification language in a life cycle based on the spiral model. It is extremely important to standardize the communication in relation to the requirement specification between the demanding team and the research center team. In this sense, the use of UML diagramming is suggestible during the phases.

The use case diagrams are a graphic view that indicates system functions specifically, and the sceneries involved for each case must be sent, intending to reduce risks. The class diagrams, where the system entities can be seen, must be sent as well. Too many details are not needed, being recommendable a greater details on critical systems only, if existent. Other diagrams are considered important to maintain communication between distant teams that are the activities (what need to be done by the participants?), collaboration (which actor must do certain activity?), packages and architecture diagrams. Sceneries are also used to help process development, specially in the phase of requirement definition.

Each phase of the project (requirement definition, project, production, evaluation, transition and liberation) will have a set of activities described next. Besides the presentation of these activities associated to each phase of the proposed model, it will also be presented the set of basic activities to control the execution of the software development project in the physically distributed environment.
5.1. Phase 1 – Requirement definition

The process begins in this phase, where the collecting of user’s needs is done, looking forward to process formalization and risk reducing. This phase includes characteristics of the conception phase (figure 3) of the unified process life cycle. A team elaborates a written plan of software development and documents requirements for project planning. The demanding team creates a document called development intention, which must have an overview of the user’s point of view. Sceneries are used to ease the communication process between participants. Everything must be distributed for the rest of the team after conceptual studies approval, confirming the possibility of project elaboration (milestone 1).

The following are the activities to be done in this phase:
- Written plan of software development;
- Requirement documentation for project planning;
- Conceptual studies approval.

5.2. Phase 2 – Project

The instantiation and the organizational definition of the project are done in this phase. This phase includes characteristics of the conception phase (figure 3) of the unified process life cycle. The demanding team, together with coordinators (from both locations), assigns the project manager, who will assemble the rest of the team, with the right resources allocation. The team must be aware of the project understanding and combined with the project manager elaborate the preliminary project specification. This specification must define the domain logic of the application for project and its scope (characteristics and restrictions), that is, the project purpose, detailing the development interaction document.

Use case diagrams are used in this phase for system dynamic requirement documentation. They are presented on user level, which describe how users interact with the system. After that, the development is approved (milestone 2), and the project proceeds.

The following activities are done in this phase:
- Project Instantiation;
- Organization definition:
  - Project manager assignment;
  - Team assemble;
  - Resource allocation;
  - Technological aspects;
- Team aware of the project understanding;
- Project preliminary specification;
- Development approval;
- Preliminary analysis:
  - Requirement specification;
  - Project scope definition;
  - Objectives definition;
  - Preliminary architecture;
- Domain conceptual modeling;
- Schedule definition.

5.3. Phase 3 – Production

This is the phase where project/analysis, involving all teams, and codification, involving development team most of the time, are done. This phase includes characteristics of the conception phase (figure 3) of the unified process life cycle. A preliminary production is developed, and then the analysis of this production is done. Architecture definition, component identification, data modeling and elaboration of complementary diagrams are some activities done by integrants of several teams together with the project manager. Programming, documenting, code reviewing, development tests and module integration are specific activities of the development team.

Use case diagrams, in developer level (that described how components are going to interact), class, architecture, packages and collaboration diagrams are developed and distributed among several teams for updating and conservation of the existent communication in the project. The production approval (milestone 3) finishes this phase.

The following activities will be done in this phase:
- Analysis/Project;
- Preliminary Production;
- Analysis of the preliminary production;
- Architecture Definition;
- Component Identification;
- Data modeling;
- Elaboration of complementary diagrams.
- Coding:
  - Programming activities;
  - Documentation;
  - Code Reviewing;
  - Development test;
  - Module integration.
- Production approval.

5.4. Phase 4 - Evaluation

This phase brings the maximization of positive results and minimization of negative events consequences. This phase includes characteristics of
the conception phase (figure 3) of the unified process life cycle. The greatest participant of this phase is the quality team, with test plan elaboration, and performing specific and general tests of the project.

Tests diagrams are used in this phase, illustrating the performing of class, subsystem, package, components and script tests. Use case diagrams appear in this phase as well. The iteration between the project manager and the quality team aims the approval of the main changes done (milestone 4), looking forward to proceed to the transition phase.

The following activities will be done in this phase:
- Maximization of positive results;
- Minimization of negative events consequences;
- Test plan elaboration;
- Performing of specific tests;
- General test of the application by the quality analysts;
- Changes approval.

5.5. Phase 5 – Transition

Improvements and final adjustments are part of this phase. This phase includes characteristics of the conception phase (figure 3) of the unified process life cycle. The project manager, together with the demanding team, must finish the project within the schedule. The project is delivered to the client and after the approval, final version is concluded. The system administrators do the installation and liberation phase begins.

The following activities will be done in this phase:
- Improvements and final adjustments;
- Project must be within the schedule;
- Client approval;
- Final version conclusion;
- Installer creation;
- Installation done by the system administrators.

5.6. Phase 6 – Integration

This phase is the connection between the developed software and the business processes to which it will give support. To perform the integration process, it is necessary to obtain the reliability in the system, data integrity and get project performance. Fulfilling these conditions, the project is given to operation and to production support, followed of a critical analysis of the project, as much by the demanding team as by the development team and users.

The following activities will be done in this phase:
- Obtain the reliability in the system;
- Keep data integrity;
- Obtain project performance;
- Give it to operation and to production support.

6. Final Considerations

It is easy to find software analysts and engineers without the adequate knowledge to comprehend the general requirements to manage a software project, and there is no way to know how long would it take to acquire such knowledge. In an environment where physical distribution exists (user, analysts, developers etc) the problems in the software development process are seen more critically. This makes planning, size estimation, to change and to produce software more difficult. Problems such as requirement specification, test process and the communication between participants etc can be highlighted.

The approaches mentioned are important for the definition of a project management model toward a physically distributed environment. As much in the procedural approach as in the object-oriented one, it shows the relationship among phases and how are they interconnected to each other throughout the project. The proposed model resulted from the necessity of finding answers for a critical problem presented in the work environment we’re involved in. This problem is centered in the communication difficulties derived from the physical and cultural distance of user and developer group.

For future work, the intention is to specify and implement a support software prototype for this model. This work is considered a relevant contribution to the project management area, since it tries to aggregate important study contributions of the area ([CAN 98], [SCH 00]) aiming to propose a customized model to reply to the research issue proposed.

A great growth potential in this research path can be identified, where high points involve a stable partnership between academy and industry, creating unique experimentation and learning conditions, derived from a positive synergy between partners.

7. Bibliography


