TOWARDS A MODEL OF SOFTWARE DEVELOPMENT PROCESS FOR A PHYSICALLY DISTRIBUTED ENVIRONMENT

Minimizing communication difficulties and adding planning and evaluation view to the software development life cycle

Rafael Prikladnicki
Pontifical Catholic University of Rio Grande do Sul, Computer Science School
Master of Science Student
Porto Alegre, Rio Grande do Sul, Brasil, 90619-900
Email: rprik@inf.pucrs.br

Jorge Luis Nicolas Audy, Ph.D.
Pontifical Catholic University of Rio Grande do Sul, Computer Science School
Professor at Computer Science School
Porto Alegre, Rio Grande do Sul, Brasil, 90619-900
Email: audy@inf.pucrs.br

Keywords: Distributed Team-based Software Development, E-Business, Evaluation, Information Systems, Planning and Software Development Process.

Abstract: The goal of this article is to present the first results of a study looking forward to propose a model of software development for a physically distributed environment (United States and Brazil). Thus, the objective is to combine empirical and theoretical knowledge in the software development area, aiming to minimize communication problems found in this specific environment. As result, the article proposes the addition of two new phases in the software development process, planning and evaluation, based on Unified Process and the UML language. The purpose is to demonstrate through illustrations and discussions, how these two phases are engaged in the process as a whole and what they represent. Some points open to discussion, which are themes for future researches, are identified, aiming to propose a model of software development for a physically distributed environment. The empirical basis of the study involves the E-Business Research Center, a partnership of Dell Computers with PUCRS.

1. INTRODUCTION

Nowadays, software has become more and more important for modern society [13]. For this reason, the processes of software development have increasingly become more important in the search for quality, productivity growth and technological evolution in the organizations. This area had a huge progress, creating a great concentration of efforts towards establishing a behavior guided by quality and continuous improvement. The basis for such is the existence of a software development process adapted to a specific reality, which allows continuous improvement every project.

Besides, the integration between the corporative strategic planning and the planning in the area of IS is considered to be an important matter in the competitiveness and success in the organizations recently. Particularly in the area of IS, the planning process can and must be the guider of the technological environment and of identifying and prioritizing the systems to be developed.

Several studies have placed the software development process as a technical activity dissociated from the organizational sphere [17] and [26]. Many efforts can be identified in order to define, establish and spread out information systems development approaches which assure more adequacy to the user' specifications and requirements. These approaches must be aligned to the business strategy of the company. There are several kinds of systems development. Different kinds of problems and challenges have different characteristics that require different kinds of approaches. The major challenge is to select, to adapt and to integrate these approaches, in accordance to the characteristics found in a certain environment [14] and [17]. Amongst these approaches it is pointed out the one object-oriented, and more specifically, the UML (Unified Modeling Language) and the RUP (Rational Unified Process).

This research has the objective to propose a model of software development for a physically distributed environment. This model must contemplate the interfaces for planning activities as initial phase for the software development process, in the context of an organization that have off-shore areas of software development, distributed worldwide (India and Brazil in this case). The purpose is also to analyze the possibility of inserting an evaluation phase as a final phase of the software development process, enabling continuous learning and changes resultant from this evaluation process that will help to improve the proposed model. Besides, it presents the results and an analysis of a study case, in developing specific software for the e-business website of the organization being studied, and the resultant contributions that generated the necessary elements for the model being researched.

The research issue can be defined as: Which are the necessary characteristics to the software development process, in order to adequate it to a physically distributed environment?

1.1 Software Development Environments

Big organizations are scattering more and more their software development processes around the world, intending to increase productivity, lessen of costs and gain of quality. In this context, countries such India and Brazil emerge as potential candidates to host these worldwide software development centers. In relation to Brazil, the specific legislation in the computing area stimulates companies, through tax deduction, to invest resources on applied research projects in agreement with Universities and local Research Centers. Several organizations of high importance in the computing area are taking advantage of these incentives and redirecting theirs efforts towards the creation of software development centers in order to attend to the demand of the organization in its headquarters, typically USA and Europe [20].

In this context, it points out the arising of a new problem class in the software development process that involves the cultural differences and the physical distances between the participants of the process. This way, the traditional problems related to the development process, strongly centered in the requirement specification phases and the system analysis, get more critical surroundings. The way to solve these problems is centered on the adoption of more formal and defined specification and development process languages. Verification and Certification models of the maturity level of the software development process, such as CMM (Capability Maturity Model), have become more and more useful and important in order to the contractors organizations to have a minimal guarantee about the quality of the utilized process by the partners system development organizations or laboratories. The era of monolithic and informal development approaches is ending. The systems developers have conscience of the existence of multiple forms of specifying and developing the systems [17]. New technologies and information systems types, such as the expert systems, inference and rule machines, neural networks and genetic algorithm require different development approaches.

In other words, if there are different ways to specify a system [14], even considering a same development paradigm (i.e., Object Oriented), why not to choose the best matching of the usage of these approaches for a certain type of problem or environment? Why to stuck up to a model or single and inflexible methodology, if a matching of approached, techniques and tools can create a more precise, adequate, economic and elegant result? The research is led in the software engineering area in order to create its own software development model, based on a coherent conceptual framework, the object oriented (OO). The scientific community accepts the providing of models, granted by the conceptual basis of OO. These models have been assumed as requirements of which extensions and adjustments are proposed to help answering the proposed research question.

This paper has the following structure: the item 2 presents the theoretical basis used; the item 3 discuss the research method being used; the item 4 describes the case study, with the experience lived and its analysis and the item 5 presents the preliminary model proposed as a result of this analysis. Finally, item 6 presents the final considerations, guiding towards future studies and research limitations.

2. THEORETICAL BASIS

2.1 IS Planning

Information Systems Planning (ISP) is the process of identifying an application portfolio that is based on a computer to support the organization’s business plans and to help the concretization of the organizational objectives [16]. Many researches have been guided to improve the IS planning process. The studies in this area have investigated the IS and business strategy alignment [3], [12, [22] and [27]; the identification of the opportunities to take competitive advantage using IT [11], [19] and [24]; the plans applicability of plans [8], [9], [15] e [22]; and the relation with the software development process [25] and [26].

Several authors review the nature of the strategic planning in the Information System area. This way, studies aiming to evaluate if the IS strategy must be planned separately or if it is a continuous process in which the new ideas come up throughout the operations. The starting point of these authors is that the software development process must be a consequence of planning in the IS area, in which the formulation of the strategy is seen as a learning process. This way, in this study, the objective is to incorporate the planning process view as a first stage of a set of software development projects.

2.2 Unified Modeling Language

The Unified Modeling Language (UML) is the successor of the object modeling languages, unifying Booch (Grady Booch), OMT (James Rumbaugh) and OOSE (Ivar Jacobson), according to [24].

It is by definition a language used to specify, visualize and document the artifacts of a system based object oriented, along the development process. Besides, it is a tentative to standardize the analysis and design artifacts: semantic models, syntactic notation and diagrams [21]. It provides a standardized form to plan a system in such a way, enclosing the business-oriented processes and functionalities of the system, as well as the artifacts constructed during the system development. For being just a language, UML is only part of a software development methodology [13].

2.3 Rational Unified Process

The Rational Unified Process is a software engineering process [13] that supports the incremental and interactive life cycle, based on a spiral model for software development. It is a generic methodology based on processes that provides organized scenarios, which assign tasks and responsibilities inside an organization. One of the most interesting aspects of RUP is the fact that it can be adapted and extended according to the organization's requirement and settings. Its main authors are the ones in charged of drawing attention towards it. The objective is to increase the software production quality that faces final user's requirements [13], [21].

An overview of the proposed process by the RUP can be seen in figure 1, where there are two dimensions:

- ***An horizontal dimension***, representing the dynamic part of the process, that is, time.
- ***A vertical dimension***, representing the static part of the process, which are the components of the process.

The time (horizontal dimension) is characterized by 4 well-defined phases and its interactions. Each phase has an objective, as described next:

- ***Conception Phase***: it has as objective to specify the vision of the project.
- ***Elaboration Phase***: the objective here is to specify the resources needed and to plan the required activities, detailing what must be done.
- ***Construction Phase***: it has as objective the construction of the project according to what has been specified in the earlier phases.
- ***Transition Phase***: the objective is to execute the tasks related to the product delivery to the respective users.

Each phase is concluded with well-defined *milestones*, that is, a critical point where there must be evaluated if the objective of this phase has been achieved.

On the other hand, the components of the process (vertical dimension) are the activities involved along the period, where the effort spent varies according to the development phase.

The components of the process can be defined as:

- ***Business modeling***: systems objectives in the business environment;
- ***Manage requirements***: management of the functional and non-functional demands set;
- ***Project and Analysis***: detailed description of how the systems will be built based on the iterations done so far;

- ***Implementation***: building of the system source code having support in the documentation generated in the previous iterations;
- ***Test***: quality control of the process and systems created;
- ***Deployment***: all activities related to the delivery of the system to the final user.

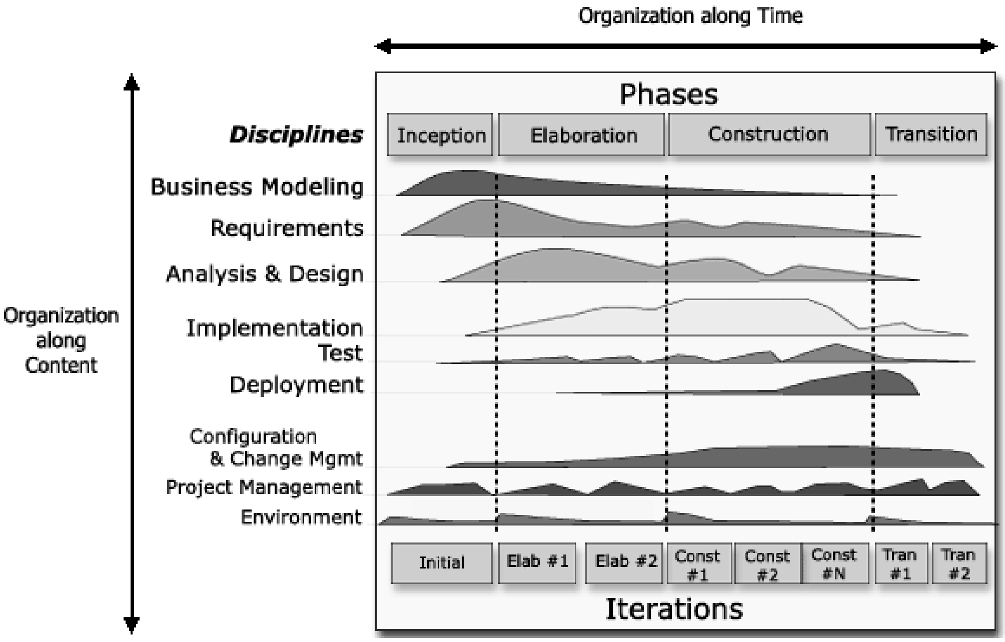

Figure 1: Rational Unified Process [10].

Different companies have utilized the RUP since the end of 1999, applying it to several applications contexts (e-business, corporative systems, etc.) and different project areas. [10] points out that this methodology is versatile and has more than 50% of its usage focused on e-business and planning.

3. RESEARCH METHOD

Although the ample theoretical review developed, there is no idea that the problem proposed had been approached under the same perspective. This way, this research is characterized as a study mostly explanatory, since the main research method was the case study.

In this study, of explanatory nature, it is possible to justify the usage of qualitative methods since it involves the study of the system development process in its real context, with description and the comprehension of the art state in those situations where practice precedes theory [28]. In relation to the study nature, the explanatory research has as main objective to develop, clarify and change ideas and concepts, focusing on the formulation of new theories, models and researchable hypothesis on former studies. According to this author, the explanatory research is constituted sometimes of a more ample investigation in the first stage. This is exactly the situation of our research, developed in a Software Development Research Center in the e-business area located in Porto Alegre, Brazil, resulted of an agreement between DELL Computers, through Latin-America Online Division, Austin, Texas and the PUCRS University (Pontifical Catholic University of Rio Grande do Sul).

The study case method is adopted as proposed by [28] and it has been developed in a system development context for the DELL Computers E-business Web Site. Due to the high level of involvement of the authors with the object of analysis, the case study report follows the recommendations proposed by [22] and used in several studies in the area [1], [2] e [3], characterized as lived through experiences. The object of analysis has been specifically the approaches and the problems related to software development in a physically distributed environment, and the communication between the developer team and the project manager (in Porto Alegre, Brazil), and the client team and the program manager (in Austin, TX, United States).

Since it is a qualitative research, limitations of this type of research, especially concerning the number of organizations studied must be clear, astricting the obtained results generalization.

4. CASE STUDY

4.1 Characterization of the Organization

The DELL/PUCRS Research Center (RC) in e-business was created in Porto Alegre in the second semester of 2000, being created using incentives of the Brazilian Law on Information Technology that stimulates companies located in the country to invest part of their earnings on research and development institutions providing tax exemption on manufactured products (IPI). The RC aims to perform scientific research and technological development in the e-business area, making usage of professional of DELL and the excellence of the PUCRS researchers on areas that may solve problems and fulfill needs existent nowadays in systems of this area. This way, the objective is to foment the scientific research and promote technological development from scientific results, generating advanced prototypes, integrating systems and creating new electronic commerce systems specifically for the e-business site of DELL Computers.

In relation to its activities, the Research Center has entire autonomy to decide how to work, always using a democratic mechanism, creating an important contribution for the course of the work.

In this context, this study tried to analyze, throughout the six first months of activities (from April to September of 2001), what was the existent software development process and how the project execution has occurred during this period. Finally, an analysis has been started, from where some conclusions were taken pursuing a definitive model, which will be described next.

4.2 Lived Experience

During the 6 months of monitoring, it was possible to verify some well-defined periods in relation to the application of software development processes. Figure 2 shows the phases of the empirical researches described:

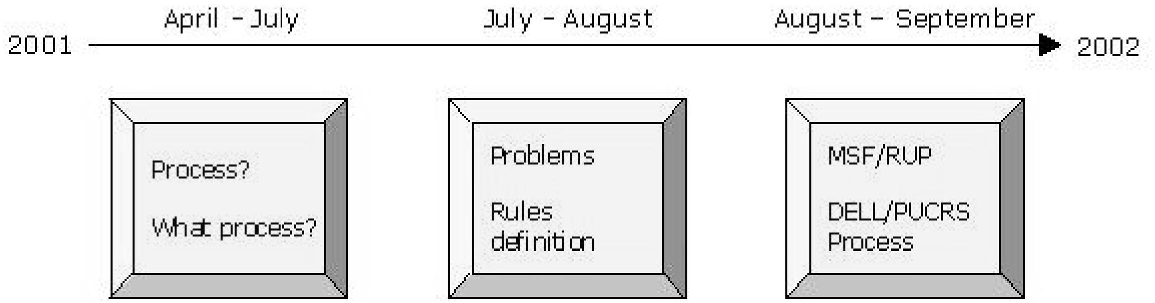

Figure 2 – Phases of the empirical research

During the initial period (April to July of 2001) there wasn't a software development process, and the people part of the Research Center developed in the way they believe to be correct, and according to the previous experienced brought over from other organizations. Thus, there wasn't a well-defined process, only some processes that were naturally implemented depending on the project demand. Besides, the communication in the USA wasn't formalized either, causing several problems of requirement specification, deadlines, and documentation generated.

In the following period (July to August of 2001), the first projects development has shown the existence of several problems in the development process, concerning problems within the project team, having difficulties to develop something based on a process, and communication problems with the user team, because besides the distance, there wasn't any formalization, what made development harder. Thus, some rules that would start to be applied were defined during this period. Rules that would certainly improve the work, but didn't contemplate the entire development process. From this moment on, people started to discuss the software development process that should urgently be implemented and become of everyone's knowledge. A study to do so was started.

In the final period of the empirical research (August to September of 2001), after theoretical studies, the option was to implement an existent software development process, based on Unified Modeling Language (UML) and on the Unified Process. Activities started to succeed, even applying the exsitent software development process used in great organizations. Yet there were still some existent problems, which characteristics were really alike: there was some discomfort in some moments for this physically distributed environment.

4.3 Case Study Analysis

After 6 months of monitoring, even implementing the existent software development process, based on RUP an its four well-defined phases (Conception, Elaboration, Construction, Transition), it was possible to verify the existence of some specific problems in a distributed environment that were not supported by the existent models studied. As result of a conjoined analysis among project participants and coordinators, the following problems were found (Chart 1):

Number	Problem
#1	The project did not start well because there was no formal beginning and the requirements were not completely communicated to the project team.
#2	Contacts of each project started were not known.
#3	There wasn't an overview of all projects being developed.
#4	There wasn't an overview of future project planning, regarding the time period.
#5	There wasn't a feedback from the users, USA's project manager and Research Center's project manager regarding the project evaluation, in the user point of view or the development one, according to the development process previously set.
#6	There wasn't a mechanism that would allow process improvement every project, because there wasn't a way of evaluating the projects done.

Chart 1 – Problems that were found after 6 months of monitoring.

These analyses involve a number of meetings where the perceptions of the technical team in Brazil, of the technical team in USA and of the Brazilian and American coordinators were registered. The systematizations of the results of these meetings, of the kind of brainstorm, having as focus the development process evaluation, led to the following lesson for study (Chart 2):

Number	Lesson
#1	It is necessary to standardize the sending and reception process of functional and non-functional requirements between USA and Brazil.
#2	It is necessary greater planning and communication to indicate the contacts of each project.
#3	It is necessary greater project entrance planning, its deadline and necessary team for performing certain project.
#4	Necessity of a formal planning phase aiming to align USA clients and Porto Alegre team's perspectives.
#5	Necessity of an evaluation phase where it is possible to appraise the project entirely, by the final user and by its execution team. Necessity of evaluating the project and the development process aiming for improvements for future projects.
#6	To create metric collection mechanisms throughout software process that would be part of the final evaluation of the development process. These data would later be appraised and to have their usage considered for future projects.

Chart 2 – Lessons learned.

Based on problems found and lessons learned (Chart 1 and 2) in the case study, a deep theoretical process was started in the identified themes. AS result, a blank was found in the models of software development when talking about a physically distributed environment when development teams are off-shore in relation to users and clients, as well as part of the technical interface.

Thus, the intention is to propose a model, which fulfills these blanks in the models of software development. This model is described next.

5. PROPOSED MODEL

The intention was to use RUP as basis for the software development process, but it appear the necessity of inserting the four RUP phases in a more adequate and consistent model to the challenges of the research area. In this sense, two other phases were added to the process; one in the beginning phase (planning) and other in the ending phase (evaluation).

Considering that the RUP has well-defined phases and several authors mention this subject [5], [13] and [10], only the new phases of the software development process will be characterized next. Figure 3 shows the preliminary proposed model, incorporating planning and evaluation phases.

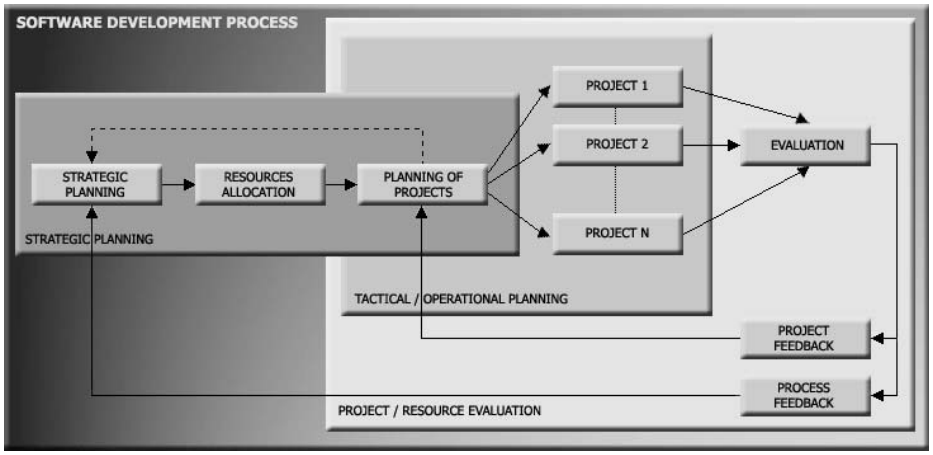

Figure 3 – Proposed Model

These new two phases are directly related with the lessons learned from the case study developed. Lessons #1, #2, #3, and #4 drive the planning phase, and lessons #5 and #6 drive the evaluation phase. The outcome from case study and the strong theoretical review process gave empirical and theoretical basis to the proposed model (figure 3).

As research progress, future studies will contemplate customizations and descriptions of changes proposed for one of RUP phases, aiming consolidation of the physically distributed software development environment. In a third moment, development of software tools will start in order to support the use of the proposed environment and methodologies.

5.1 Planning Phase

As we expand the focus of the software development process, and try to adopt a more strategic position in relation to the process, we identify the planning stage as the first one to take place. Thus, the planning stage basically involves the definition of the strategies, which will lead the development process entirely, throughout the time.

It is possible to consider the planning stage as a former cycle of many project cycles derived from the planning process. Figure 3 shows this relation.

There are two cycles of planning in the context of an organization that owns a software development center that is physically far away from its head office, which must keep frequent communication with each other. The first one lies on the Strategic Planning, with direct participation of the organizations involved in the RC. The process is internally guided in each organization by the coordinators assigned by their institutions, and the coordinators are responsible for the Strategic Alignment among the perspectives and goals of each organization in relation to the objectives, strategies, and policies of the RC. The second cycle involves the tactical strategic planning in the scope of the RC, which will be detailed next. The overlapping of the two planning cycles occurs exactly when the projects are planned. The result is a document, which will identify and set priorities to the different projects to be worked on.

In relation to the identified projects, we may consider that many of them involve specific software development projects, which will follow their own development methodology, and will be based on the RUP generic proposal, with 4 well-defined steps (conception, elaboration, construction and transition). The reference model for the development process is the spiral model, with all of its implications, especially in relation to the interaction between its phases and resources.

The proposed approach reinforces the spiral model, since it creates effective feedback conditions, whether to end the first cycle (strategic planning), as to end the second one (tactical/operational planning).

The tactical planning stage is coordinators final responsibility (approval) in each partner institution (or operational unity geographically distributed of the same organization). The negotiations and contacts related to these definitions can occur between coordinators and people responsible for the project entrance at the same time.

The operational planning involves specifically the software development project administration centered in the general coordination of the work among the collaborators, interfaces among teams, communication, and contacts with the clients and conflict solving. The final responsibility of this phase belong the project manager, the one responsible for all the documentation requested for the project.

5.2 Evaluation Phase

In relation to evaluation of software development process, two main aspects must be pointed out:

- Evaluation of the software development process, under the responsibility of the project manager and his or her team (process evaluation);
- Evaluation of the developed software, regarding final users, administrators and the project manager himself or herself and his or her team.

In a distributed environment, it is extremely important to have a possible procedure of data collection and measurement of the work developed. It is important since it aims development process and final product improvement. Being a physically distributed environment, it is particularly important to know what people physically distant from the software development process think about the process, since they will be the users or clients of the generated results. Thus, this will only be possible after the creation of some procedures made for this kind of environment. It was possible to verify in RUP, and some variants, such as Microsoft Solution Framework (MSF) [18], that both didn't have clear evaluation procedures, for the process as well for the final product. In relation to the evaluation mechanisms they refer to software tests and conformity verification with specific quality standards of the software engineering area (CMM, SEI, ISO norms, etc.). Thus, there isn't a specific phase where a complete evaluation relating the aspects cited previously can be done. For this reason, the idea was to propose in this model the creation of a phase called evaluation, where the main objective is evaluates the software developed in both aspects cited earlier, aiming to focus the process in the relation among the business process that will be supported by the system, the system itself and the technology used.

This phase would be done after the transition phase, set by the RUP. That is, the conclusion of the software development process is done after the evaluation phase, establishing the feedback mechanism for the planning cycles previously set (first and second cycle).

The choice was the creation of an explicit phase because since the context is a physically distributed environment, the simple fact of being an independent phase makes the evaluation process more adequate and valid. It will only occur when the software project is delivered. Besides, it is easier to implement it being a phase since the efforts can be concentrated just in this activity.

This phase has two objectives:

- Collect information about the project's life cycle, measuring the satisfaction of the stakeholders involved in order to determinate high and low points of the process.
- Evaluate the course of the project as a whole, having as evaluators basically the project managers and his or her team.

For each objective a specific technique is used, which will be explained next:

In relation to the first objective, forms are submitted for clients, evaluating several aspects of the project. The quantification of the result of this evaluation is used as reference to trace high and low points presented during the project's life cycle. The collection of information that will be used for project analysis is done through forms destined to specific functions performed by the client, being able to be, for instance, the project manager and the final user. The forms must be fulfilled and turned in to the project manager, who will repass it to his or her evaluation group.

In relation to the second objective, several meetings are held in the end of the project to discuss how the project was developed, personal relationship, adopted decisory process, participation level, problems found. As result, the information collected will be the basis for the formation of a knowledge base that will potentially act as a continuous learning vector, leading to better results in the next projects. The use of brainstorming and group techniques seem to be adequate in this aspect.

6. FINAL CONSIDERATIONS

The proposed model resulted from the necessity of finding answers to a critical problem presented in a physically distributed environment these days. This problem is centered in the communication difficulties derived from the physical and cultural distance between both user's and developer's group. This way, the answer was searched in the formalization of a communication and specification process, having well-defined roles for the participants, besides universally accepted standards in the software engineering area, such as Object-oriented, UML language and the Unified Process of software development and requirement engineering.

Based on the theoretical and practical experience involved in the process, as much in the academic as in the business sphere, the major objective is to develop a ser of approaches and models focus on the characteristics of a geographically distributed organization, with significant cultural and linguistics differences, with high technical and departmental level though.

The preliminary results being found lead us to trustful indications that the search for greater formalism in the development process and the selective use of international standards give us plenty conditions to overcome these cultural and linguistics differences, especially in the creation of a complete software development model for a physically distributed environment, which is the focus of this paper. The insertion of the planning and evaluation phases proposed in this article, allow an effective experience of the spiral life cycle, diminishing risks associated to software development process, particularly to the aspects related to deadlines and system scope.

The model proposed in this article establishes the conceptual basis, with consistent empirical sources, to deepen the study aiming to overcome challenges caused by the distributed environments of software development. The insertion of these new phases (planning and evaluation) in the software development process, as part of the process, establishes a research field and customization of the current model of software development, that present significant gaps to attend organization's demands. A great growth potential can be identified in this research line, where high points involve a stable partnership between industry and academy, creating unique experimentation and learning opportunities, derived from a positive synergy between partners.

REFERENCES

[1] Audy, J. and Lederer, A. Seven principles of organizational learning in information systems planning: preliminary findings from a case study. *Proceedings of AMCIS*, Long Beach, CA, 2000.

[2] Audy, J. L. N. *Strategic Planning Model of Information Systems: contributions of decision process and organizational learning (in Portuguese)*. Ph. D. Tesis, PPGA – UFRGS, Porto Alegre, BR, 2001.

[3] Brodbeck, Angela F. *Strategic Alignment among Business Plans and Information Technology Plans (in Portuguese).* Ph. D. Tesis, PPGA – UFRGS, Porto Alegre, BR, 2001.

[4] Boar, B. *The Art of Strategic Planning for Information Technology*. New York, John Wiley and Sons, 1993.

[5] Booch, G., Rumbaugh, J., Jacobson, I. *Unified Modeling Language (in Spanish).* Addison Wesley, 1999.

[6] Cantor, M., 1998. *Object-oriented management with UML.*

[7] Fowler, M., Scott, K., 2000. *UML distilled: applying the standard object modeling language*, 2nd edition.

[8] Gottschalk, P. Strategic information system planning: the implementation challenge. *Proceedings of AIS 97*, Indianapolis, USA, 1997.

[9] Gottschalk, P. *Content characteristics of formal information technology strategy as implementation predictors*. Ph.D. Dissertation. Henley Management College, Brunel University, 1998.

[10] Jacobson, I., Booch, G., Rumbaugh, J., 1999. *The unified software development process.*

[11] Kearns, G. and Lederer, A. Alignment of IS plans with business plan: the impact on competitive advantage. *Proceedings of AIS 97*, Indianapolis, USA, 1997.

[12] King, W.R. How effective is your IS planning?. *Long Range Planning* (21:2), 1988.

[13] Kruchten, P., 2001. *The Rational Unified Process an Introduction*, 2nd edition.

[14] Larman, C., 2000. *Using UML and standards: an introduction to the analysis and to the object oriented project (in Portuguese).*

[15] Lederer, A. and Salmeia, H. Toward a theory of strategic information system planning. *Journal of Strategic Information System*, 1996.

[16] Lederer, A.L. and Sethi, V. The implementation of strategic information system planning methodologies. *MIS Quarterly* (12:3), September 1988.

[17] Martin, J. and O'Dell, J. Object Oriented Methods: A Foundation. Prentice-Hall, NJ, 1998.

[18] Microsoft Solution Framework, online in http://www.microsoft.com/msf.

[19] Porter, M. E. and Millar, V. E. How Information gives you competitive advantage. *Harvard Business Review*, Boston, Jul/1995.

[20] Prikladnicki, R., Peres, F., Audy, J., Móra, M. C., Perdigoto, A. Requirements specification model in a software development process inside a physically distributed environment. *Proceedings of ICEIS 2002, Ciudad Real, Spain, 2002.*

[21] Quatrani, T., 2001. *Visual Modeling with Rational rose and UML (in Portuguese).*

[22] Reich, B. and Benbasat, I. Measuring the linkage between business and information technology objectives. *MIS Quarterly*, March 1996.

[23] Reponen, T. The Role of Learning in Information System Planning and Implementation. In: GALLIERS, H. e BAETS, R. *Information Technology and Organizational Transformation.* Chichester, England, John Wiley and Sons, 1998.

[24] Service, R. and Boockholdt, J. Employing information systems for competitive advantage. *Proceedings of AMCIS*, Long Beach, CA, 2000.

[25] Sprague, R.H. and McNurlin, B.C. *Information Systems Management in Practice.* Canada, Prentice Hall, 1999.

[26] Valacich, J., George, J. e Hoffer, J. Essentials of Systems Analysis e Design. Prentice-Hall, NJ, 2001.

[27] Venkatraman, N. Beyond Outsourcing: Managing IT Resources as a Value Center. *Sloan Management Review*, spring 1997.

[28] Yin, R. K. Case study research: design and methods, Sage. 1994.