TOWARDS A PROCESS TO INFORMATION SYSTEM DEVELOPMENT WITH DISTRIBUTED TEAMS

Gislaine Camila Lapasini Leal
Production Engineering Department, State University of Maringá, Maringá, Brazil

César Alberto da Silva, Elisa Hatsue Moriya Huzita, Tania Fatima Calvi Tait
Computer Department, State University of Maringá, Maringá, Brazil

Keywords: Distributed software development, Development process.

Abstract: The software engineering area offers support to manage the information systems development process. Software engineering processes have artifacts, roles and activities well defined, allowing adjustments in specific situations. However, these processes are generally oriented to development of coallocated projects. They are not taken into account peculiarities in regard to coordination, control and communication, when the development is with distributed teams. The purpose of this paper is to contribute to the Software Engineering area presenting a comparative analysis of some development processes from the perspective of development with distributed teams. Additionally, it points to the need to process definition that includes the peculiarities of this approach to software development.

1 INTRODUCTION

Nowadays, we live in a time that is increasingly growing need for high quality software. There it is necessary to seek ways to improve the development process, whether through the use of tools or artifacts generated. An example would be the use of formal methods to describe the specification, design and test (Abdurazik et al., 2000).

The software process is defined as an ordered set of activities for the management, development and maintenance of software, and should be aligned with the organizational conditions (Fuggetta, 2000). The software as part of an information systems brings organizational elements besides technical such as: human resources or teams, organizational culture among others. The processes of software development are, as noted in the current literature, focused on the development of projects co-located.

The Distributed Software Development (DSD) is a recent approach to software development that meets the demands of globalization. This new strategy added to the software development challenges related to cultural differences, geographic dispersion, coordination and control, communication and team spirit. The distributed teams face to challenges related to technical and non-technical factors. The technical factors relate to the problems of network connectivity and the differences between the environments of development and testing. The non-technical factors, have issues of trust, communication, conflict and cultural differences (Damian, 2002). So the challenges of communication, coordination and control generated by this approach, arising from the physical distances and time, demand for studies that present the guidelines for defining a process of software development that is suitable for distributed teams.

This paper presents an analysis of some software development process from the perspective of distributed teams, and raises some important points that should be considered in this development approach. Thus, some guidelines are established for a distributed software development process.

2 RESEARCH METHODOLOGY

The research conducted in this work is characterized as being qualitative basic and exploratory and was prepared in two steps. In the first step was made an initial review of the literature, aiming to obtain a consistent theoretical basis for continued study and to

Camila Lapasini Leal G., Alberto da Silva C., Hatsue Moriya Huzita E. and Fatima Calvi Tait T. (2010).
TOWARDS A PROCESS TO INFORMATION SYSTEM DEVELOPMENT WITH DISTRIBUTED TEAMS.
In *Proceedings of the 12th International Conference on Enterprise Information Systems - Information Systems Analysis and Specification*, pages 493-496
DOI: 10.5220/0002909804930496

have view the state of the art. The second step consisted in the systematic review following the standard of protocol defined in (Biolchini et al., 2005).

A systematic review was conducted to expand the coverage of the initial review of the literature. The main objective was the identification of works that deal with software processes used by distributed teams. In addition, the review also aimed to identify the needs of a development process for this development approach.

A systematic review allowed to identify the software development process which has been used by distributed teams. Most studies selected were related to problems (cultural, geographic dispersion, coordination and communication) found in the use of distributed development, emphasizing the need to define a process that incorporates the features of this new development approach.

The results provide valuable information for defining a process that meets the needs of distributed software development.

3 CHALLENGES IN DISTRIBUTED SOFTWARE DEVELOPMENT

This section describes the criteria and analysis of: *Rational Unified Process* (RUP) (Kruchten, 1999), *Extreme Programming* (XP) (Beck and Andres, 2004), *Agile Unified Process* (AUP)(Ambler, 2008) , *Extended Workbench Model* (Paulish, 2007), (Avritzer, Hasli and Paulish, 2007) and *Local Agile Game-based Process* (LAGPRO) (Ribeiro, Czekster and Webber, 2006). After then, some guidelines for defining a process considering the challenges of communication, coordination and control of the distributed development approach are also presented (Subsection 3.2).

3.1 Criteria and Process Analysis

The RUP and XP processes were chosen because they are widely used, and is a traditional methodology and an agile, respectively. Rocha et. al (Rocha et al., 2008) presents an experience report about the adaptation of RUP for small distributed development teams.

The AUP was chosen because it is a simplification of RUP and uses the concepts of agile methods. The Extended Workbench Model and LAGPRO are used in DSD context, according (Paulish, 2007), (Ribeiro, Czekster and Webber, 2006), and (Urdangarin, 2008).

A set of benchmarks based on the characteristics of distributed software development identified in (Urdangarin, 2008) (criteria A - G), the basic elements of a software process (criteria H - J) and the type of methodology (criterion K and L) were considered to a comparison among above mentioned software development process. They are described as follow:

- **(**A) Diversity: identifies whether the process provides mechanisms for control, coordination and communication of differences (cultural and language) among the teams involved in the project.
- **(**B) Requirements: analyzes if the process has a formal specification of requirements, aiming at minimize the problems of ambiguities in the artifacts and so reduce the communication.
- **(**C) Centered: identifies if the process has centralized control of project activities.
- **(**D) Decentralized: identifies if the process is characterized by decentralized control of project activities.
- **(**E) Monitoring: identifies if the process support an adequate control of project developed by remote times.
- **(**F) Collaboration: checks if there are tools to support collaboration among teams as a way of sharing information.
- **(**G) Trust: identifies if the process offers support to establish trust among members of different teams.
- **(**H) Roles: identifies if the process has criteria to define roles.
- **(**I) Artifacts: identifies if are adequately defined all artifacts produced.
- **(**J) Activities: identifies if the process defines the activities to be carried out in each phase.
- **(**K) Traditional methodology: verify if the process focus is a traditional methodology of software development.
- **(**L) Agile methodology: verify if the process has characteristics of an agile methodology.

Table 1 shows the comparison of the processes in relation to the criteria above mentioned. Next is presented an analysis on them.

There is not a process that includes criteria related to the DSD (criteria A - G) and present the basic elements (roles, artifacts and activities). Thus, there is a lack of a systematic process that incorporates the features of DSD. Therefore, it is necessary that defined activities, artifacts and roles required at all stages of development using the distributed approach were defined.

Table 1: Comparative analysis of processes.

Process/ Criteria	A	B	C	D	E	F	G	H	I	J	K	L
RUP				x				x	x	x	x	
XP				x	x	x	x	x	x	x		x
AUP				x				x	x	x		x
Extended Workbench Model		x	x	x	x			x	x			x
LAGPRO		x	x	x	x	x	x	x	x			x

RUP, XP and AUP define the minimum set of elements that make up a process. However, do not have procedures to address issues related with DSD problems, which were represented by the criteria A to G.

Although the Extended Workbench and LAGPRO have been developed for use in project developed in distributed way, they haven't mechanisms to show the diversity, they talking in account only requirements formal specification. Extended Wokbench does not mention the factors related to collaboration and trust. Furthermore, we didn't found reports about the artifacts generated. In LAGPRO case, the activities are not identified. Thus, it is possible to show that is necessary a process that meets these new demands generated by the distributed approach and presents a definition of process elements.

Audy et al. (Audy and Prikladnicki, 2008), point out that in a distributed software development environment is fundamental a common development process for the team, because a methodology helps directly in sync, providing all team members a common tasks and activities naming also, and a common expectations for all individual involved in the process.

According to Rocha et al. (Rocha et al., 2008), when the context is Distributed Development, the scenario changes if compared with to traditional software development, because the variables and risks increase. So if there is not a good methodology for the development process, the project will have a good chance to not correspond to the initial planning. In their first insight into the study and use of DSD, Rocha et al. (Rocha et al., 2008) emphasized the need of more adequate process, since, based on research carried out was not easy to identify process models for the DSD. So, it reinforces the need for a development process that effectively provide adequate support to characteristics of DSD.

3.2 Guidelines for Software Distributed Development Process

From the analysis described in Section 3.1 was defined a set of guidelines for the definition of a software process that meets the needs of distributed development, listed as follow:

- adopt hybrid management (centralized-decentralized) related with to the activities control;
- encourage communication between development and integration test teams offering artifacts with relevant information for them;
- use formal specification of tests to mitigate the problems of ambiguity and to reduce the need for communication;
- using a component-based architecture for develop in a distribution way or perform the phase distribution, such as development and testing, thereby reducing communication among teams;
- support for awareness, so that the people involved are aware of place and responsibilities assigned to each team member;
- adopt language with low learning curve and semantics to provide information to developers, for example, the use of UML because its is known in both academia and industry;
- provide continuous integration to reduce casual problems of integration;
- have iterative development and frequent deliveries to provide greater visibility of project managers;
- define an infrastructure that enables collaboration, documentation control and artifact versions control; develop and apply test suites; and establish method to control documentation;
- define a language to formalize the process and interaction among the teams; and,
- define of a leader to foster trust and commitment among members.

4 CONCLUSIONS

The physical distribution of teams increases the problems of development management. Cultural differences, language, time zone among other things, increase the complexity of communication, coordination and control during software development.

Using a standardized process for planning, offers support to project managers by allowing you to set plans in accordance with the standards and quality procedures of the organization (Berger, 2003). Moreover, the adoption an appropriate process to the characteristics of organizations, provides a consistent and predictable development (Kruchten, 1999).

Thus, we analyzed the technical, methods and software processes found in literature that contributed

to the understanding of its main activities, actors and artifacts. Thus, the definition of a specific process for DSD provides a common language to establish roles and activities, enabling a better understanding of the business domain terms and goal of the project, despite cultural differences and organizational structure. Therefore, a process must support distributed development improving communication, reducing misunderstanding and uncertainty arising from cultural differences and language, reducing redundant activities and excessive work.

The main contribution of this work is the set of guidelines presented in Section 3.2, which aims to reduce the problems of communication, coordination and control in software development with distributed teams. As future work, we have: the definition of a development process that meets the criteria presented in Section 3.1, defining the phases, roles, activities and artifacts. This will reduce the problems of ambiguity and understanding of the artifacts, setting a single representation for all involved teams, therefore allowing them to manage and use collaborative tools, and thus encourage the establishment of trust among team members.

REFERENCES

Abdurazik A.; Ammann P.; Ding W.; Offutt, J. Evaluation of Three Specification-Based Testing Criteria. In 6th IEEE International Conference on Complex Computer Systems (ICECCS'00), Tokyo, Japan, pp. 179-187, (2000).

Ambler S. The agile unified process (AUP), Capturado em http://www.ambysoft.com/unifiedprocess/agileUP.html, August of (2008).

Audy J.; Prikladnicki R. Desenvolvimento Distribuído de Software: Desenvolvimento de software com equipes distribuídas. Rio de Janeiro: Elsevier, (2008).

Avritzer A.; Hasli NG W.; Paulish D. Process Investigations for the Global Studio Project Version 3.0. In: International Conference on Global Software Engineering (ICGSE 2007), IEEE Computer Society, Washington, DC, pp. 247-251, (2007).

Beck, K.; Andres, C. Extreme Programming Explained: Embrace Change, Cambridge: Addison Wesley Professional, 224p, (2004).

Berger P. Instanciação de Processos de Software em Ambientes Configurados na Estação TABA. *Dissertação (Mestrado)* - COPPE/UFRJ, Rio de Janeiro, RJ, (2003).

Biolchini, J. and Mian, P. and Natali, A. and Travassos, G. Systematic review in software engineering: Relevance and utility. *Technical Report*. COPPE/UFRJ, (2005).

Fuggetta A. Software Process: A Roadmap. In: *International Conference on Software Engineering, Proceedings of the Conference on The Future of Software Engineering*, (2000).

Damian, D. Workshop on Global Software Development. In: *International Conference on Software Engineering (ICSE'02)*, IEEE Computer Society, USA, pp. 19-25, (2002).

Kruchten, P. The Rational Unified Process - An Introduction. Addison-Wesley, Boston, USA, (1999).

Paulish, D. J. Scalable Distributed Organizations for Ultra-Large-Scale Software. In: *International Workshop on Software Technologies for Ultra-Large-Scale Systems (ULS'07)*, pp. 4-4, (2007).

Ribeiro, M.B.; Czekster, R. M.; Webber, T. Improving Productivity of Local Software Development Teams in a Global Software Development Environment. In: *International Conference on Global Software Engineering, IEEE Computer Society*, pp. 253-254, (2006).

Rocha, R.; Arcoverde, D.; Brito, R.; Arôxa, B.; Costa, C.; Silva, F. Q. B.; Albuquerque, J.; Meira, S. R. L. Uma Experiência na Adaptação do RUP em Pequenas Equipes de Desenvolvimento Distribuído. In: *II Workshop de Desenvolvimento Distribuído de Software*, Campinas, SP, pp. 81-90, (2008).

Urdangarin, R. G. Uma Investigação sobre o Uso de Práticas Extreme Programming no Desenvolvimento Global de Software. *Dissertação (Mestrado)* - Faculdade de Informática, PUCRS, Porto Alegre, RS, (2008).