Globally Distributed Software Process Engineering

L. García-Borgoñon

Research, Development and Technological Services
Area Instituto Tecnológico de Aragón
Zaragoza, Spain
laurag@ita.es

Advisor: MJ Escalona

IWT2 Group
Universidad de Sevilla
Sevilla, Spain
mjescalona@us.es

Abstract— **Software processes is becoming a more addressed issue in software development companies every day. These processes are defined regardless of the environment in which they run. To incorporate aspects of that environment is essential, especially if referring to GSE. Despite this fact, the process itself should not be necessary modified. This paper provides a first draft of a research focused on software process definition, modeling, implementation and evaluation in a GSE environment, so as to facilitate the information exchange through a hierarchical process that does not involve modification of specific processes.**

Keywords-software process, software process modeling, process improvement, method engineering, model driven engineering

I. INTRODUCTION

The practice of Global Software Development (GSD) has significantly evolved in recent years and now represents an important alternative in IT companies [1]. It has some implications which differ from traditional software development, with its benefits but also challenges, mainly due to the distance factor (geographic, temporal, linguistic and socio-cultural) [2]. Coordination and communication distances are key aspects in Global Software Engineering (GSE) [3], and in many cases, this coordination begins with the software process.

Most organizations have their own software processes. If we propose a scenario where two or more companies work in liaison to develop a software product from their own independent processes, one of them usually imposes its process, or even the development of specific procedures, to the remaining companies. This is an effort made by at least one of the organizations. If the same scenario is considered, but without domination from any of the parts, there is a need for establishing a way to coordinate and communicate those processes.

It is also possible that some of these organizations meet a quality standard such as CMMI (Capability Maturity Model Integration) [4]. However, this new scenario does not ensure the adequacy of that standard, what can be seen as an additional challenge to the existing ones.

Besides, it can be observed that some of the organizations may follow agile methods widely used in GSE environments. Nevertheless, this should not always imply that a company needs to apply such methods, since the company may be using some others suitable for its necessities.

This is not unique for GSE as it may occur even if companies are collocated. Otherwise, if we propose a scenario GSE, the problem will get worse as including aspects related to the above mentioned challenges will be essential.

Furthermore, one of the main aspects in a GSD environment is the existence of tools to support it. [5]. They must be connected, led and orchestrated by software processes.

The aim of our study is to establish mechanisms and tools that enable this software processes coordination and collaboration in a global development environment (GSE).

II. RELATED WORK

Software process, defined as "the set of partially ordered process steps, with sets of related artifacts, human or computerized resources, organizational structures and constraints, intended to produce and maintain the requested software deliverables" [6] is a widely discussed topic in engineering of processes. It highlights the varied factors affecting software processes and their differences with the typical production processes summed up in dependence on people, and therefore, communication, coordination and cooperation [7].

After a first-generation software process modeling languages with a formal approach similar to the traditional production processes [8] [9] [10] [11], in recent years, a second generation of UML-based language have been developed. These benefit from the issues derived from the MDE (Model Driven Engineering) [12] [13] [14] [15]. The best choice differs depending on the context and according to several factors, such as evaluation of the execution capacity, semantic richness, graphical representation, modularity and support tools existence.

Proposals for the dynamic adaptation in software processes execution time, have also been carried out due to the flexibility this processes demand [16].

However, these previous studies do not consider issues caused by the GSD.

Regarding support tools, there are individual solutions to solve specific processes within a GSD working environment, but there is no connection among these tools unless you are working in the commercial suite environment that is integrated within their proprietary platform [17].

Finally, there are also many approaches and case studies on successful implementation of quality standards such as CMMI [2], and multi-model and multi-site approaches [18], but none of them cope with the specific case of companies that collaborate and maintain different processes whose combination complies with the standard.

III. RESEARCH OBJECTIVE

The summary of the problem and related studies shows that, although there are some previous software process engineering approaches, none of them incorporates the characteristics of GSD in terms of collaboration among organizations.

The main goal of this research is to incorporate aspects of GSD to software process modeling as well as develop a framework that allows its implementation and sets a link to support tools. In addition, this would enable further evaluation according to quality standards within the context of collaboration among organizations.

Another goal is to define, within the process modeling field, hierarchical processes, which will address GSE aspects without a complete change of the software processes themselves.

In order to achieve this goal, the following questions will be addressed:

- Which concepts should be incorporated into the existing software processes modeling languages to consider the characteristics of GSD? And, how are they related to Agile methods and self-organization?

- How is it possible to enact and execute a software process modeling in a GSD context?

- What are the integration requirements for process support tools in a GSD environment, so that these are processes-driven and aspects of self-organization can be incorporated?

- How can we ensure compliance with quality standards in a GSD environment as so described?

IV. RESEARCH METHOD

The present work follows the Action Research method [19], an iterative method where the experimental settings are modified according to the observation results. It consists of the following steps:

1. Planning: Identify and specify the problems and the suitable lines of work to define the proposal.

2. Action: Experiment with the proposal results.

3. Observation: Gather information from the experiments to obtain a quantitative analysis of the application framework.

4. Reflection: Analyze the results in order to achieve new knowledge to improve the initial proposal.

The paper intends to follow this method to answer each of the research questions. While the scope of this work seems very ambitious, our approach deals with leading the global problem to identify potential open issues in detail, prioritizing them according to the results obtained.

V. EARLY RESULTS

This research started with a systematic bibliographical review related to software processes and processes in GSD, to identify its main proposals and weaknesses and define the requirements the framework should meet.

Modeling software processes is found in two organizations, CMMI level 2 and level 3 respectively. They are based on a specific meta-model, SPEM 2.0 [20], in order to know the collaboration requirements that are not currently supported but can be incorporated into the newly developed framework.

Although additional studies are required, we made a first proposal on the support tools integration within a process environment [21].

As a result, we are currently about defining the framework attending to all the information gathered for further development and validation with case studies.

VI. EXPECTED RESULTS

As a final result of this research, a framework that allows us to answer the research questions raised in the previous section will be developed.

To answer the first two questions, the framework will allow the definition, implementation and evaluation of software processes, and will support the specific needs of the collaboration among organizations at the process level. For this purpose, (**it**) shall define the elements to be incorporated into the software process modeling in order to enable this collaboration. This represents a kind of process that has been defined in part of the research as *hierarchical processes*, and additionally incorporates implementation issues.

Moreover, to face the third question, we propose to develop an environment that helps us to support the processes execution and their project instances in the case of GSD and, more particularly, in cases of collaboration among organizations. One aspect that achieves special importance in our investigation is the definition of metrics; how they can be established and analyzed in this GSD environment to be used in decision making.

Finally, to solve the last question, we intend to incorporate process modeling aspects for their evaluation to ensure they support quality standards.

VII. EVALUATION METHOD

The results obtained assessment will be carried out through case studies into real companies.

In these case studies, different scenarios will be established to face up the research questions previously made. In this regard, we will work with companies in different countries with different methodologies, some of which will incorporate agile methods, whereas others will have quality certifications in

different standards and models, so we can evaluate whether our proposals meet the needs of GSE in the field of software processes engineering.

To this end, measures and acceptance metrics in the adaptation process from every organizations view will be established.

More studies will be needed in order to know what indicators allow for further validation of the approach, and which will be medium-term analyzed and established.

REFERENCES

REFERENCES

[1] Agerfalk, P.J., et al. Benefits of Global Software Development: The Known and Unknown, in International Conference on Software Process, ICSP 2008. 2008. Leipzig, Germany: Springer Berlin/Heidelberg, p.1-9.

[2] Casey, V. Leveraging or Exploiting Cultural Difference?, in International Conference on Global Software Engineering, 2009. Limerick, Ireland, p8-17.

[3] Herbsleb, J.D., Global Software Engineering: The Future of Socio-technical Coordination, in Future of Software Engineering, 2007, Minneapolis, USA, p188-198.

[4] Chrissis, M.B., et al., CMMI Guidelines for Process Integration and Product Improvement. Addison-Wesley, 2nd edition, 2007.

[5] Portillo-Rodriguez J., Vizcaino A., Tools to Support Global Software Development Processes: A Survey, in International Conference on Global Software Engineering, 2010. Princeton, NJ, USA, p13-22.

[6] Lonchamp, J., A structured conceptual and terminological framework for software process engineering, in International Conference on Software Process ICSP. 1993. Berlin, Germany. IEEE Computer Society Press, Los Alamitos.

[7] Ruiz-Gonzalez, F., Canfora, G. Software Process: Characteristics, Technology and Environments, UPGrade: The European Journal for the Informatics Professional, 2004, vol. 5, no. 5, p6-10.

[8] Sutton, S.M., Jr., Heimbigner, D., Osterweil, L.J., APPL/A: A Language for Software Process Programming, ACM Transactions on Software Engineering and Methodology, 1995, vol. 4, no. 3, p221-286.

[9] Kaiser, G.E., Barghouti, N.S., Sokolsky, M.H., Preliminary Experience with Process Modeling in the MARVEL Software Development Environment Kernel, in 23rd Annual Hawaii International Conference on Systems Sciences, 1990, p131-140.

[10] Bandinelli, S.C., Fuggetta, A., Ghezzi, C., Software Process Model Evolution in the SPADE Environment, IEEE Transactions on Software Engineering, 1993, vol. 19, no. 12, p1128-1144.

[11] Conradi, R., et al. Design, Use and Implementation of SPELL, a Language for Software Process Modelling and Evolution, in Second European Workshop Software Process Technology, 1992, Berlin, Germany, p. 167-177.

[12] Bendraou, R., Gervais M.-P.,Blanc, X.UML4SPM: An Executable Software Process Modeling Language Providing High-Level Abstractions, in 10th IEEE International Enterprise Distributed Object Computing Conference, 2006, p. 297 - 306.

[13] Bendraou, R., Gervais M.-P.,Blanc, X. UML4SPM: A UML2.0-Based Metamodel for Software Process Modeling, in ACM/IEEE Eighth MoDELS Conference, 2005, p.17-38.

[14] Bendraou, R., Jezequel, J.-M., Gervais, M.-P., Blanc, X., A Comparison of Six UML-Based Languages for Software Process Modeling, IEEE Transactions on Software Engineering, 2010, vol. 36, no. 5, p662 - 675.

[15] Bendraou, R., Jezequel, J.-M., Fleurey, F., Combining Aspect and Model-Driven Engineering Approaches for Software Process Modeling and Execution, in International Conference on Software Process, ICSP 2009. 2009. Vancouver, Canada: Springer Berlin/Heidelberg, p.148-160.

[16] Kabbaj, M., Lbath, R., Coulette, B., A Deviation Management System for Handling Software Process Enactment Evolution, in International Conference on Software Process, ICSP 2008. 2008. Leipzig, Germany: Springer Berlin/Heidelberg, p.186-197.

[17] IBM, Jazz. Available from http://jazz.net

[18] SEI publications on CMMI mappings and comparisons, http://www.sei.cmu.edu/cmmi/casestudies/mappings/comparisons.cfm

[19] Baskerville, R., Wood-Harper, T., A Critical Persepective on Action Research as a Method for Information Systems Research, Journal of Information Technology, 1996, vol. 11, p235-246.

[20] OMG, SPEM2.0 Software and Systems Process Engineering Meta-Model Specification, http://www.omg.org/spec/SPEM/2.0, 2008.

[21] Garcia-Borgoñon, L., Avanzando hacia la interoperabilidad de los sistemas de soporte a la mejora de procesos, in SEPG Latinoamerica Conference 2007, Santiago, Chile.