SPI methodology for Virtual Organizations

Paula Ventura Martins[1], Alberto Rodrigues da Silva [2]

[1] INESC-ID, CSI/Universidade do Algarve
Campus de Gambelas, Faro, Portugal
pventura@ualg.pt
[2] INESC-ID /Instituto Sup erior Técnico
Rua Alves Redol, nº 9 –1000-029 Lisboa, Portugal
alberto.silva@acm.org

Abstract. This paper discusses the importance of software process improvement in a virtual environment where several organizations are cooperatively involved in the development of a software product, each one using its own development process. The main focus of the paper is a methodology, called Process and Project Alignment Methodology, to improve the development process of a single organization based on projects knowledge. However, the authors believe that the same fundaments can be applied in a virtual organization and discuss the extension of the presented methodology to a virtual organizational context.

1 Introduction

New and emerging market conditions are the core engines driving organizations in focusing on their competences and to cooperate with others under networks called Virtual Organization (VO), where each member has its own organizational culture and, in the context of software development, each one has its specific development process. Therefore, it is necessary to create a common specification to represent different processes for each organization. After this unified process representation, it is possible to define VOs development processes. To do so, does not mean to represent in a model every detail of all organizational processes involved in the VO. Instead, the organizations that participate in the interactions only have to describe interface issues. Software development process details which are internal to an organization should not be represented at this level but must be properly encapsulated. However, the involved context requires each organization to have an approach to process management before dealing with VOs development processes.

For now, our focus is to create a methodology to define and improve an organizational development process. Since project management is the discipline that controls and monitors deviations from the original project plan and also manages all process disciplines, project management is the right way to detect changes in the project that can lead to process improvement. Considering the relationship between the process and the project, new software process improvement (SPI) approaches have to consider process and project alignment and iterative SPI performed by project teams. Currently, there seems to be a lack of support on how SPI approaches addresses the problematic about how development processes are effectively applied and improved using knowledge fromsoftware projects. Another challenge is how to control and validate important project changes that must be integrated in the process. In this paper we propose a SPI methodology based on process and project alignment that enables improvements on organizational processes. We also discuss and propose a solution that enables interactions between a virtual project and projects locally executed in different organizations. This methodology is supported by two meta-models that allow project definition based on a software process. We propose to extend these meta-models by allowing publishing and subscribing to events, and by enabling the definition of points in the project where events should be sent or received.

This paper is organized in the following sections. Section 2 presents a literature overview about alternative approaches to project management, process management and SPI. In the context of VO, we will present a brief description about VO projects related with process and project management. Section 3 discusses the problematic about SPI, process and project management. Section 4 describes the proposed methodology to support iterative SPI based on process and project alignment. In this section, we also discuss mechanisms to extend meta-models of the proposed methodology to support SPI in VOs. Finally, Section 5 presents conclusions, limitations of the methodology and future work.

2 Related Work

Process and project management is discussed by Budlong, Szulewski and Ganska [1], Climitile and Visaggio [2] and Chan and Chung [3]. But, only the AHEAD approach [4] has a fundamental feature: its support for process improvement. However this SPI solution isn't implemented in a project management context.

Considering that process and project alignment conduce to SPI activities, we present two SPI approaches discussed in the literature. Traditional SPI methods and approaches are based on final project retrospectives [5]. In these methods, there is a long time span between the problem identification and the validation of the new process. On the other hand, agile SPI approaches have a different perspective. According to agile principles [6], the project has reflections meetings in regular intervals. Cockburn proposes a reflection workshop technique [7], Dingsøyr and Hanssen have a workshop technique called postmortem review [8], whereas Salo and Abrahamsson discuss a Post Iteration Workshop (PIW) method [9].

Therefore, all these approaches have no solutions to provide project management based on a process description and also iterative SPI. The main challenge in iterative SPI is to have project changes in real time, so project management must include fast feedback from each member of the project team.

Since this paper subject is about SPI in VOs, we present a short review about interactions between processes in VOs. Various forms of process interactions types are defined in literature, which we briefly summarize: capacity sharing, chained execution, subcontracting, (extended) case transfer, loosely coupled, public to private approach [10]. The problem of process management in VOs also had been addressed by approaches using the notation of agreements and contracts, like the WISE [11] and CrossFlow [12] projects. However, these approaches do not present any concrete process management model. A detailed and interesting approach to process management in VOs has been proposed in the context of CMI project [13]. In the meantime, the problematic about effective use of the development process in virtual projects persists.

The core foundation of this paper is on how VOs manage their processes and keep projects aligned with that processes. So we describe two projects related to process and project management: (1) Intelligent Services and Tools for Concurrent Engineering (ISTforCE) and (2) Global Engineering and Manufacturing in Enterprise Networks (GLOBEMEN). ISTforCE is a European framework project, with the objective of designing a Web-based services platform through which engineers at a given design or consulting company will access the services on the Internet and collaborate in real time. It aims at creating infrastructure on which real construction companies and virtual teams of construction companies can rent and customize services on a project by project basis, and where providers of engineering services can market their products. In the ISTforCE, the authors stated that an Internet desktop system for engineers should have the following five requirements: it should be (1) open enough to integrate with other service or tools, (2) customizable to persons, (3) customizable to projects, (4) scalable, and (5) extendable [14]. Another project is the GLOBEMEN project, which aims to create IT infrastructures and related tools to support globally distributed product life cycle management, project and manufacturing management in the VO. The project focus is on VO information exchange and control on three core business processes of manufacturing industries: (1) interaction with customers and users including global product life cycle management, (2) optimization of the delivery chain through VO resource planning and (3) distributed concurrent engineering [15].

3 Problem description

Project management, process management and SPI are interrelated disciplines that contribute to successful projects. So process management and SPI must be present during the entire execution of the project, even in the initial planning. Project planning is the most important phase in project management. The effort spent in identifying the proper needs and structure for organizing and managing a project could be minimized if the initial plan is process-based. Also important is the fact that project management monitors and controls activities from all the other process disciplines, so changes in these disciplines best practices will be detected throughout the project life cycle. SPI must be performed during project time and not only in dedicated evaluation periods. Projects are dynamic systems whose associated processes must always be under improvement.

However, many organizations have their development processes described but they don't effectively apply them in their projects. The defined process is not directly matched to their projects entities because organizations don't use process knowledge in project management.

4 Process and Project Alignment Methodology

Our research proposes a methodology that allows the definition, evaluation and improvement of an organization software development process. This proposal, called a Process and Project Alignment Methodology (PPAM), allows a general vision on the current state of an organization development process, as well as project alignment with the development process. Considering the theories and concepts described in the proposed methodology, we also discuss the mechanisms necessary to use PPAM in the context of VOs.

PPAM is based in a modelling approach since process and project modelling are the techniques used to define and analyze the significant aspects of development processes and projects. The proposed architecture identifies and interrelates the concepts necessary to provide SPI based on process and project management issues. In this paper we just show the application of the meta-models used to define processes and projects, a more detailed description of the meta-models is presented in [16]. This paper focuses on software process improvement and on how these meta-models can be used to solve this problem.

This section describes the components of PPAM essential to have process and project alignment. Process and project alignment formalization has four components: (1) process modelling enables an easy way to graphically construct a process; (2) project modelling (based on a process) provides the necessary coordination facilities for process and project alignment; (3) project control and monitoring enables observing changes in the project that are candidates to SPI. Process versioning enables creating process versions based on the proposed process improvement; and (4) process assessment allows the evaluation of the benefits due to process improvements.

Considering VOs, we propose a novel approach to virtual process and projects. This is supported by the idea that a virtual process or virtual project can be considered as a cooperation of several existing process or projects of collaborative organizations. The approach is inspired by the Service Oriented Architecture. Accordingly, the proposed methodology can be used in VOs environments but under some transformations in the following subjects: (1) virtual process definition; (2) virtual project creation and (3) groups involved. We discuss these key points as we present the components of PPAM.

4.1 Process Definition Component

A process meta-model provides a set of generic concepts to describe any process. ProjectIT Process Meta-model (PIT-ProcessM) architecture defines the concepts that correspond to elementary process concepts, allowing process creation or modification [16]. Two complementary views show those static and dynamic process elements. In the static view are represented the concepts related to process disciplines, like products, activities and roles. Meta-model dynamic view is about how a process life cycle is organized, e.g., phases and iterations. Additionally, all the process elements should be associated to a particular moment of a process life cycle.

At organizational level, the organization uses PIT-ProcessM to create his process. This step requires that the organization has knowledge about his practices based on historical data or from other process management initiatives.

In the context of VOs, PIT-ProcessM has to be extended with an interface to send and receive events. The processes created as instances from PIT-ProcessM will have an interoperability layer that identifies the services provided by each organization. The virtual process must be specified considering the services provided by collaborative organizations. Each individual organization must define its own supporting development process and must agree with other organizations on approaches to interoperability.

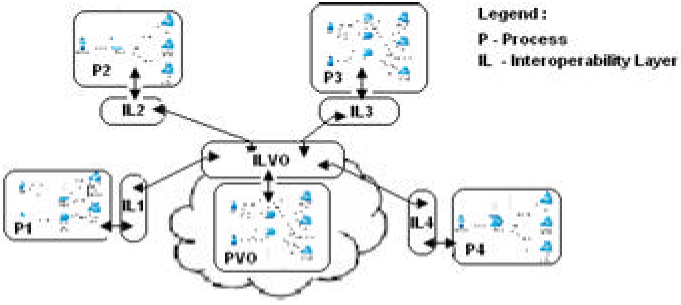

Fig. 1. VO Software Development Process

Fig. 1 presents an overview of the architecture for virtual processes. Each organization presents its own process (P1, P2, P3 and P4). The interoperability layer of each organization presents an interface to its process. The VO development process (PVO) will be defined considering the base features provided in each organization interoperability layer.

4.2 Project Definition Component

The second component (project definition considering a base process model) is essential to begin the project and consists in the project plan definition. A project is instantiated from a process, where a process represents reusable process practices at an abstract level. But in real world projects, multiple projects share the same process and are differentiated based on their specific elements, e.g. persons and the resulting relationships. Considering these differences and the process and project alignment carried out in this phase, our approach includes a ProjectIT Project Meta-model (PIT-ProjectM) to support this dependency [16].

Each organization defines their projects in alignment with a base process. In VOs, the enactment of the virtual project requires that each organization define its project and activate its interoperability layer in order to send and receive events. Event notifications will be managed by a suitable event service that is capable of filtering and correlating events, and of dispatching them to the appropriate target project.

The virtual project also needs a virtual team responsible by project activities and work products. A coordination organization must manage the virtual project. This organization must create and delegate responsibilities for the virtual project control, monitoring and evaluation. Considering that virtual project change can improve the virtual process, some members of the participant organizations must perform the roles responsible by the SPI project.

4.3 Project Coordination and Monitoring Component

The third component consists in project coordination and monitoring. Updates and extensions to the initial project plan will be registered, always considering a base process model. Considering that some changes in the project best practices can improve the base process, we introduce a process versioning meta-model. The concept of process version has the advantage that the history of evolution of the processes is recorded. Since an effective process validation is only possible after a testing period, we must provide a mechanism to keep track of the changes carried out in the process's best practices. Thus, it isn't sufficient to maintain only the current version of a process. In this context, SPI subsumes two problems: (1) process modification and (2) ensuring that projects and base process remain consistent with each other.

Versioning Process. As proposed by agile methods, SPI is an iterative initiative during the project lifetime. Our proposal includes a workshop, when a dedicated member (process group) detects changes in project best practices that are considered as candidate improvements to the process. The basic idea of the proposed methodology is not to update process in place, but to version them. When a new process is created, this is considered the creation of a first version (root version). New versions are derived from existing ones by applying one or more modification operations to the based process version. However, as we will see, versions are created in an incremental way. Therefore, we will use the concept of versions states as used by [17], three states are distinguished: transient, released and obsolete. When a root version is created, it is in transient state. In this state, a version can be updated or deleted. In order to prevent invalid processes, when a version is in state transient its not allowed to: (1) create descendent versions; (2) create projects based on that version and (4) reference the version by another version. Finally, when a version is accepted its state is changed to released. In a released state, the version can't be deleted or updated, but all the other operations are allowed. When a released version has to be modified, his state is changed to transient, but only in special conditions (no descendents versions, no projects are based on it and is not referenced by other versions). If a version becomes unused is state is changed to obsolete and it is allowed to create new projects based on that version. An obsolete version can be deleted only: if has no descendents, has no derived projects and isn't referenced by other versions. But first it has to change to transient state and then the version can be deleted.

This section presents some details about extensions to PIT-ProcessM in order to support process versioning. Fig. 2 presents the main constructs of the extended meta-model. A process includes a unique identifier (process name) and a process version tree. A process version defines a version number and it is either in state transient, released or obsolete. A process comprises one or more process versions that can be derived from another process version by applying one or more modification operations. The diagram illustrates the relationships between a process and his versions. PIT-ProcessM was updated to include modification operations applied to a process in creating a new version. Original elements from PIT-ProcessM like Phase, Iteration, Discipline, Activity, Role and WorkProduct are replaced by its versions classes. Associations between original PIT-ProcessM concepts are now performed between their version elements. Thus, the original elements have an association to its correspondent version, since each element can be used in one or more process versions.

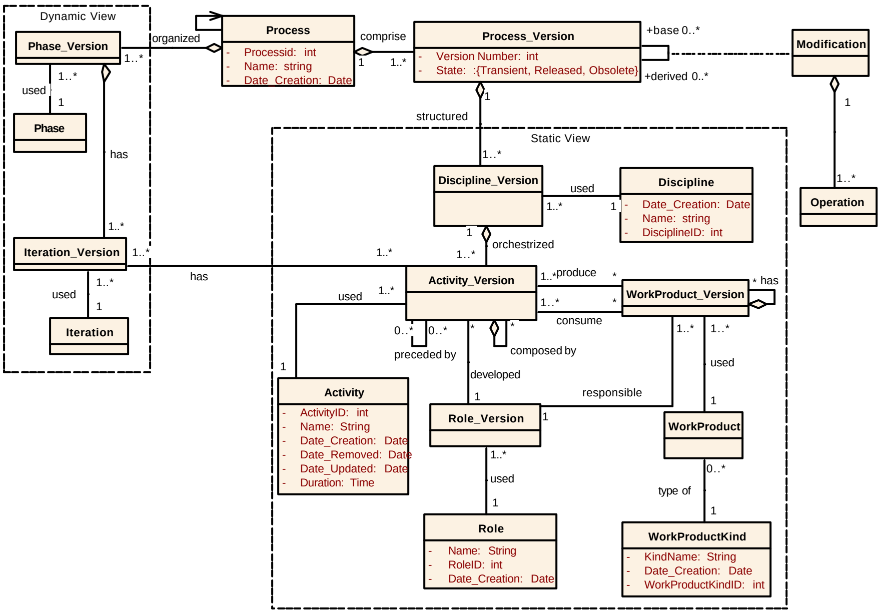

Fig. 2. PIT-ProcessM Versioning

In a virtual process, the versioning process will be performed in the same conditions as it was only one organization. However, the mechanisms used to introduce that changes can have two different sources: (1) changes in the virtual project practices or (2) changes in the processes of the participant organizations. In case of a change in the virtual project practice, the workshop proposed in PPAM has to be replaced by Internet technologies for supporting communication and collaboration (e-mail, audio or video calls, text chat, etc). But, if the new virtual process version is caused by changes in the services of involving organizations, the process group has to be responsible by the improvements in the virtual process.

Project Iterations. The groups involved in SPI consist on the software development team, project manager and the process group. The SPI method performed by these groups is realized throughout all iterations of a project, but the improvements follow a pattern that is performed in the time of two iterations (fig. 3). The SPI actions performed in these two iterations are: (1) detect improvements and create a new process version (transient state); (2) test and validate the temporary changes in the next iteration of the project.

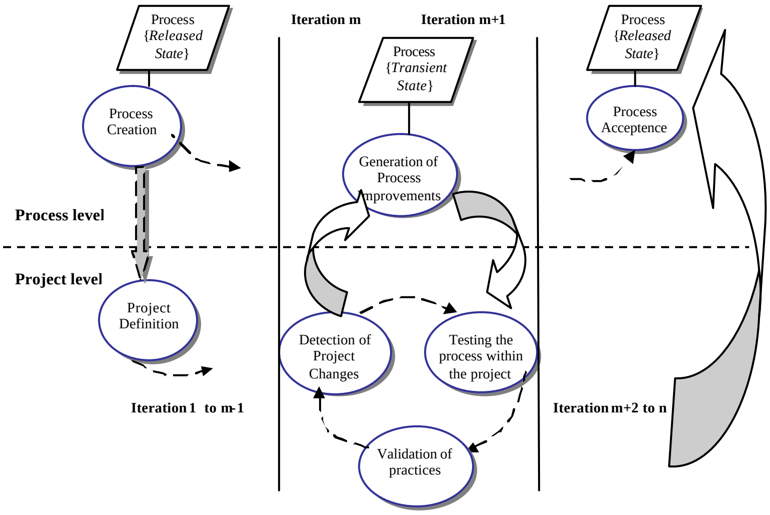

Fig. 3. Process and project alignment

In the first iteration, the project team must perform their daily work and detect situations that can lead to new practices in the project life cycle. The data collected includes positive and negative aspects found by the project team. The project manager has an important contribution in controlling the changes. At the end of this iteration, all candidate changes to improve the process are analyzed by the process group and, if necessary, a workshop is held to obtain more knowledge and present the improved process (new process version).

In the second iteration, team members get some feedback about the new practices and make notes to inform the process group. All this work will be under control of the process group. The project manager has to observe if team members are following the new proposed process. At the end of this iteration, all groups evaluate the work performed and decide if the process version is accepted. In case of success, the new process version is confirmed and the SPI method starts again. In case of failure, some new changes and improvements have been detected. The transient version will be updated and the evaluation work performed in the second iteration is repeated.

4.4 Process Improvement Assessment

In the final phase (SPI assessment), progress is evaluated throughout all process life cycle, specifying a set of improvements that can determine the process improvement itself. In the end of the project, process improvements must be analyzed in a reflection meeting. The main goal is to analyze all the improvement opportunities identified in the project and validate all the SPI actions accepted in workshops.

SPI activities can be performed successfully in many organizations. However, process managers ask themselves important questions following those activities, such as: (1) How to evaluate when a new process version meets the organization's goals?; (2) How to find if this is the appropriate development process?; (3) How will adjustments and changes affect the efficiency of the development process?; and (4) This new development process version will improve the performance of an organization?

Nowadays, project managers often lack reliable feedback from benefits of improving their development processes. In a daily basis, project managers use project management tools available in the market with the purpose of measuring and assessing software projects. However and due the fast changing environment proposed in the PPAM, the most important objectives are related with the impact on improving a software development process. Feedback information on SPI enables organizations to have control on future applications of a software process.

SPI assessment in practice can be viewed as the acquisition of data (key indicators) in a project where the new process version was applied, followed by data analysis and decisions about the further adoption of this development processes. Since project management is an important discipline in the proposed methodology, the key indicators must be those used by the project manager to analyse and evaluate a project. Normally, a project success is evaluated in terms of staff productivity, software quality, cycle time, and cost of the project. These features should be considered as key indicators to perform a SPI assessment.

5 Conclusions and Future Work

As organizations try to define their processes, they also recognize the need of continuous SPI. Even when the effort to improve them is done, organizations may fail at achieving the intended goals. Recognizing that the most critical problems occur during project activities, we strongly believe that process and project alignment can be a best-practice to get better project results and improve organizations software processes.

In this paper, we described the PPAM architecture and principles. The approach presents an innovative proposal but also includes ideas of other research initiatives in agile processes. We believe that PPAM contributes to a real process and project alignment. We present two meta-models that can support that alignment. The methodology concepts are being integrated in a project management tool (ProjectIT-Enterprise) of a research project (ProjectIT) from INESC-ID Information Systems Group [18]. In a VO domain, we discuss the extension of these meta-models to allow interactions between the virtual project and the projects executed locally in participating organizations.

As future work, our intention is to use the developed tool in real projects, to test and proof the approach. While, at the same time, new features could be found and included in the methodology. Additionally, these experiences will allow detecting new practices in the domain of organizational psychology, necessary to apply PPAM with success.

References

1. F.Budlong, P. Szulewski and R. Ganska, “Process Tailoring for Project Plan”, The Process Management Technologies Team, The Software Technology Support Center (STSC), 1996
2. A. Climitile and G. Visaggio, “Managing Software Projects by Structured Project Planning”, International Journal of Software Engineering and Knowledge Engineering, 7, 4, pp. 553-584, 1997
3. K. Chan and L. Chung, “Integrating process and project management for multi-site software development”, Annals of Software Engineering, vol. 14, pp. 115–142, 2002
4. M. Heller, A. Schleicher and B. Westfechtel, “A Management System for Evolving Development Processes”, Proceedings 7th International Conference on Integrated Design and Process Technology (IDPT 2003), Austin, Texas 2003
5. N. L. Kerth, *Project Retrospectives: A Handbook for Team Reviews*, Dorset House Publishing, April 2001
6. K. Beck et al., “Manifesto for Agile Software Development”, http://www.agilemanifesto.org/principles.html, 2006
7. A. Cockburn, *Crystal Clear: a Human Powered Methodology for Small Teams*, Addison Wesley, November 2004
8. T. Dingsøyr and G. K. Hanssen, "Extending Agile Methods: Postmortem Reviews as Extended Feedback", 4th International Workshop on Learning Software Organizations (LSO'02), Chicago, Illinois, USA, 2002, pp. 4-12
9. O. Salo and P. Abrahamsson, “Integrating Agile Software Development and Software Process Improvement: a Longitudinal Case Study”, International Symposium on Empirical Software Engineering 2005 (ISESE 2005), Noosa Heads, Australia, 17-18 November 2005
10. W.M.P. van der Aalst, “Loosely coupled inter-organizational workflows: modeling and analyzing workflows crossing organizational boundaries”, Information and Management 37, pp. 67–75, 2000
11. G. Alonso, C. Hagen, and A. Lazcano, “Processes in electronic commerce”, in ICDCS Workshop on Electronic Commerce and Web-Based Applications (ICDCS 99), May 1999
12. P. Grefen, K. Aberer, Y. Hoffner and H. Ludwig, “CrossFlow: Cross-Organizational Workflow Management in Dynamic Virtual Enterprises”, International Journal of Computer Systems Science & Engineering, Vol. 15, No. 5, pp. 277-290, 2000
13. D. Georgakopoulos, H. Shuster, A. Cichocki and D. Baker, “Managing process and service fusion in virtual enterprises”, Information Systems, 24(6), pp. 429-456, September 1999
14. T. Cerovsek and Z. Turk, “Prototype Internet desktop for engineers”, Product and process modelling in building and construction: proceedings of the third European Conference on Product and Process Modelling in the Building and Related Industries, Lisbon, Portugal, September 2000
15. J. Laitinen, M. Ollus and M. Hannus, “Global Engineering and Manufacturing in Enterprise Networks GLOBEMEN”, Proceedings of the Third European Conference on Product and Process Modelling in the building and related industries (ECPPM2000), Lisbon, Portugal, September 2000
16. P. V. Martins and A. R. Silva, “PIT-P2M: ProjectIT Process and Project Meta-model”, Proceedings of the OTM Workshop: MIOS+INTEROP 2005, Lecture Notes in Computer Science, Volume 3762, Agia Napa, Cyprus, pp. 516-525, October/November 2005
17. M. E. Loomis, “Object Versioning”, Journal of Object-Oriented Programming, January 1992
18. A. R. Silva, “O programa de Investigação Project-IT”, version 1.0, October 2004