ProPAM: SPI Based on Process and Project Alignment

Paula Ventura Martins, INESC-ID, CSI/Universidade do Algarve, Campus de Gambelas, 8005-139 Faro, Portugal; E-mail: pventura@ualg.pt
Alberto Rodrigues da Silva, INESC-ID/Instituto Superior Técnico, Rua Alves Redol, n° 9, 1000-029 Lisboa, Portugal; E-mail: alberto.silva@acm.org

ABSTRACT

Software Process Improvement is one of the main software development challenges. Unfortunately, process descriptions generally do not correspond to the processes actually performed during software development projects. They just represent high-level plans and do not contain the information necessary in a software project. This lack of alignment between the process and project is caused by processes that are unrelated to project activities and failure in detecting project changes to improve the process. Process and project alignment is essential to really find out how process management is important to achieve an organization's strategic objectives. Considering this alignment, this paper presents a software process improvement methodology designed by Process and Project Alignment Methodology (ProPAM).

1. INTRODUCTION

Organizational software process improvement (SPI) is a challenge to organizations to continually improve the quality and productivity of software and to keep up their competitiveness [1]. However, there has been limited success for many SPI efforts. Recent reports concluded that 70% of organizations attempting to adopt the CMM (Capability Maturity Model) failed in achieving the intended goals [2].

Although organizations try to define their process improvement program and get a certification in traditional SPI approaches (e.g. CMM [3], CMMI [4], SPICE [5], and Bootstrap [6]), there is a consensus that software development environments are changing constantly and team members have no obligation to sustain original SPI activities in face of difficulties. The agile software development manifesto contains a principle that supports this idea: "at regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly" [7].

Process modeling community in general base the research work on the assumption that an explicit process representation is the starting point for process understanding, improvement and communication, project teams are communicating in terms of budget overruns, patches for bug fixes and transaction monitors [8]. Other research communities, like the ones that study computer supported co-operative work (CSCW), argue that software development is a creative work with strong co-operation aspects and does not benefit from static process descriptions. So, process information must be combined dynamically with project specific information to create a detailed plan that includes information from all process disciplines with cost, schedule and quality requirements. Since project management is the discipline that controls and monitors deviations from the original project plan and also controls all of the activities from other process disciplines, it is the best way to detect changes in the project that can improve the process. Considering the dependency between project plan and process elements, new SPI approaches have to consider process and project alignment and iterative SPI performed by project team members. Process and project alignment is defined as the degree to which the project goals and plans support and are supported by the process practices. Moreover, it involves a real match between process practices and projects activities, products and actors. However, several modifications in a project can cause misalignments with the development process. These modifications can be management innovations or changes in the way the activities are executed. Furthermore, a modification may regard not only the considered activity, product or actor but it can also affect other elements having a dependence relation with the modified one.

The contribution of this paper is to define not only the process, but also to propose a mechanism to process evolution based on the changing needs of the development organization. This paper proposes a methodology - Process and Project Alignment Methodology (ProPAM) - based on process and project alignment to be applied during SPI projects for detecting misalignment between projects and supporting processes and identifying the process elements to be changed for restoring the alignment.

This paper is organized in the following sections. Section 2 discusses literature on software process modeling, process and project management alignment and traditional and agile approaches to SPI and. Section 3 and 4 presents the meta-models to support process definition and further instantiation of the project. Section 5 briefly sketches the architecture of the proposed methodology to support process and project alignment in iterative (traditional and agile) SPI approaches and also presents details about the process versioning meta-model. Finally, Section 6 presents conclusions and future work.

2. RELATED WORK

Current research on software development processes intends to define the process elements that constitute good practices, leaving implementation and enactment of the process to organizations. Curtis, Kellner and Over discussed some approaches using process modeling to support process improvement, software project management and Process-Centered Software Engineering Environments (PCSEEs) [9].

The Software Process Management System (SPMS) development identified and addressed the need for process models to be reusable, to support multiple views, to recognize process, product and human interactions to support process changes during development projects, and to support historical recording of the process over long periods of time [10]. In the domain of change management, the Problem Tracking System (PTS) is used to track errors and manage change request for the WIS (Wohnungswirtschaftliche Information System), a system build in a process oriented way to support all business processes from the area of house constructing and administration [8]. The Endeavors system is a flexible environment that allows users to create and evolve processes while a project is in progress [11]. Although Endeavors supports most of the features in process definition languages and modification of the process, some problems arise about process coordination and can lead to chaotic and disorganized development processes. The BORE tool and methodology extends the experience factory concept [12] through rule-based process tailoring, support for process modeling and enactment and case-based organizational learning facilities [13]. The AHEAD system also supports the management, versioning and modeling of development processes and provides an integrated set of tools for evolving both process definitions and projects [14].

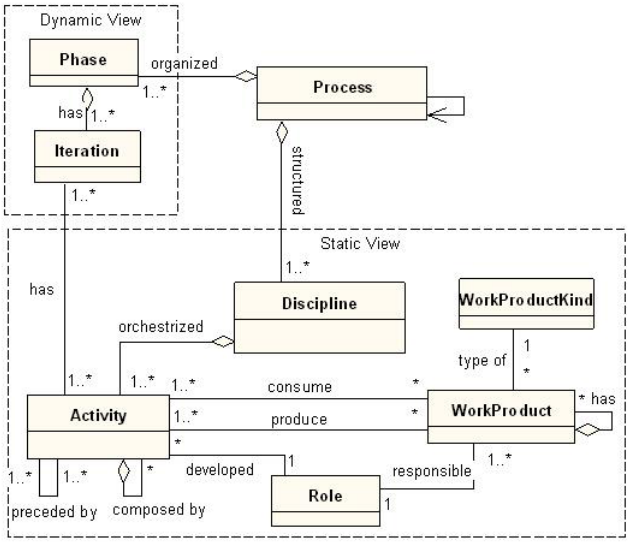
3. PROCESS MODELLING

A process meta-model provides a set of generic concepts to describe any process. Figure 1 presents the ProjectIT Process Meta-model (PIT-ProcessM) that is developed to provide a formal notation for specifying process elements involved in the software process. The meta-model consists of two complementary views: static view and dynamic view.

Static View

The static view represents formal process concepts like disciplines, products, activities and roles.

Figure 1.ProjectIT process meta-model (PIT-ProcessM)



Activity: the atomic unit of work. In operational terms, an activity can be the smallest unit of work, but also can be used to refer to a set of related activities (hierarchy of activities).

Discipline: an organizational unit to group activities according to a common "theme".

Role: describes in an abstract form the set of skills and responsibilities associated with the execution of one or more activities.

WorkProduct: correspond to typical software development objects which are produced and consumed by activities (e.g. design document, source code, test cases, etc) and they have a responsible role. The workproducts must belong to a document type (WorkProductKind).

Dynamic View

Dynamic view of the meta-model represents time and introduces concepts to describe the process lifecycle in terms of goals, pre-conditions and post-conditions and to allow the decomposition of the process lifecycle into phases and iterations.

Phase: Software development work is structured in several stages, called phases. Phases consist of a certain number of iterations and are executed with a series of milestones.

Iteration: Iterations are workflows with minor milestones.

4. PROJECT MODELLING

A project is instantiated from a process, where the process represents reusable process practices at an abstract level. While the process represents the best practices in software development but has no information about timing and resource allocation, the project must specify exactly *who* must do *what* and *when*. But in real world projects, multiple projects share the same process and are differentiated based on their specific characteristics, e.g. actors (`Person`), schedule, deadlines, resources, etc. Considering these differences that are important in a project management perspective, our approach presents a specific project meta-model presented in figure 2, ProjectIT Project Model (PIT-ProjectM).

5. PROPAM METHODOLOGY

In this section we describe the concepts behind the implementation of a process and project management system to support SPI. Figure 3 shows the relation between the four steps in the ProPAM methodology: (1) Process Creation; (2) Project Definition; (3) Project monitoring and control and (4) Process Assessment.

Process Creation

In ProPAM, a process is created as an instance of PIT-ProcessM. A process is defined by a set of phases which are divided in several iterations. Disciplines and respective activities, workproducts and roles define the space of possible choices for projects within a given process practices. Activities can be defined in an activity/sub-activities relationship represented in a hierarchical work breakdown (represented in the meta-model trough the reflexive association "`composed by`"). Activity sequences are defined through the reflexive association designed by "`preceded by`", which are useful to create a project schedule.

Project Definition

In ProPAM, processes are used as a template for creating projects. The project consists of some project elements that are instances of elements from the base process. Not all of the process elements need to be included in the project, but all project elements must be an instance of a process element. An exception is possible if an element is created as result of a change proposed by team members or process group. When a process element is assigned to a project, creating a new project element, all the information in the process element is copied to the project element, including a copy of all the associated elements. As an example, when an `ActivityProject` is created, associated `WorkProductProjects` and `RoleProjects` are also introduced in the project.

Figure 2.ProjectIT project meta-model (PIT-ProjectM)

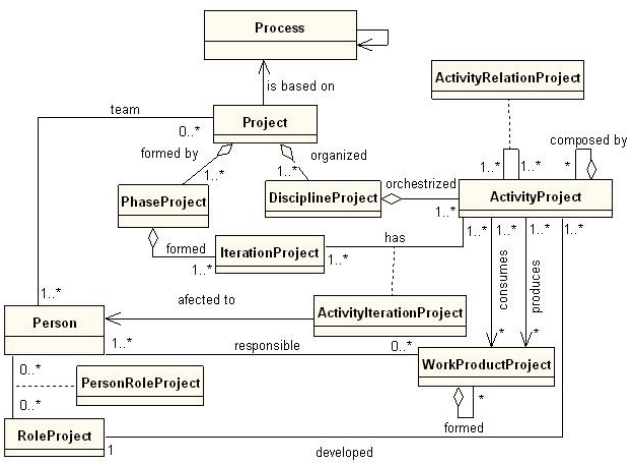
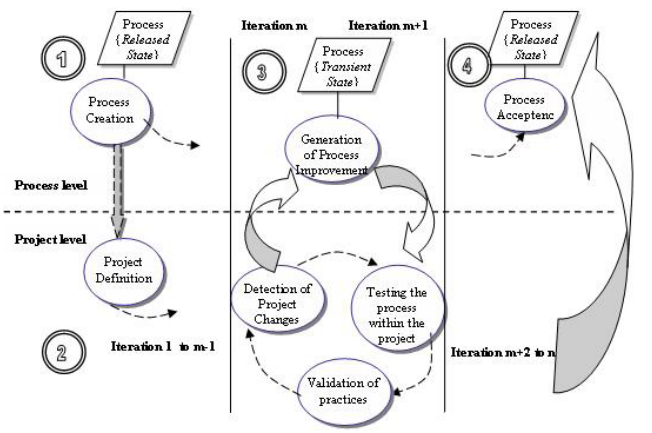


Figure 3.Process and project alignment methodology



In this methodology, after instantiating a process, the initial result is a project plan. This plan represents the initial step to start the project. Projects have certain administrative characteristics like schedule, milestones and deadline, resources, and structure i.e. phases and iterations of the project that will performed based on this initial plan. Team members are assigned to the project and gather information that is useful for the project, like workproducts, from the perspective of their current role.

Project Monitoring and Control

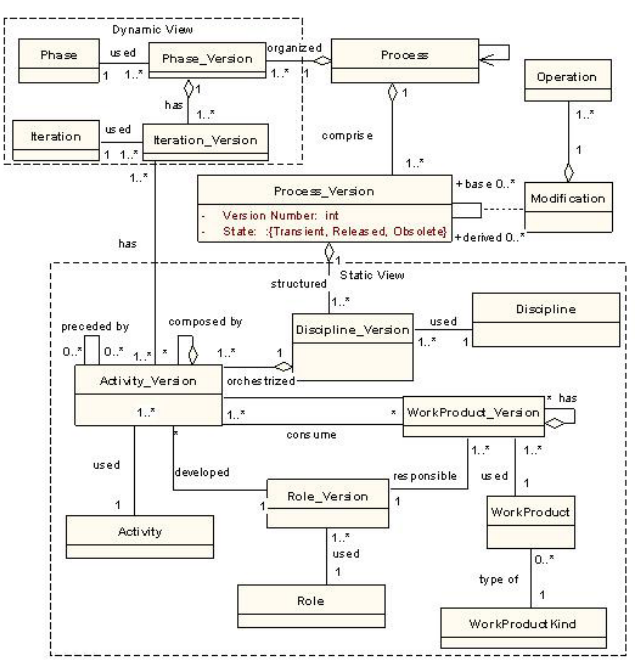
The third step consists in project monitoring and control. Updates and extensions to the initial project plan will be registered, always considering a based process model. Although most project elements are an instance of process elements, project team members have the liberty to create entities specific to a project. These changes are detected through the SPI actions performed by the process group. When a new process or new version is introduced, a validation phase is needed for monitoring their fitness and performance in the whole process. Thus, SPI actions subsume two problems: (1) process modification and (2) ensuring that projects and base process remain aligned with each other.

The project monitoring and control step is composed by two different tasks. The first task (Project Iterations) is about getting knowledge through project change candidates to improve the process and is repeated in all project iterations and phases. The second task (Process Versioning) subsumes that changes are accepted and that it is necessary to keep historical information about several process versions. The versioning schema introduces some changes in the PIT-ProcessM meta-model. The meta-model is extended to support the versioning of process elements (Figure 4).

Process Assessment

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. Prospects for success in executing and improving software process activities rise significantly when decisions can be based on quantitative information which can only be obtained by observing and measuring the products, projects and resources involved. But as complex as software development is, there are potentially so many things to measure against organizations visions and plans. In ProPAM, process assessment occurs at two distinct levels: (1) project level and (2) process level.

Figure 4.Process version classification



At project level, 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. Key indicators used to evaluate projects success are normally: staff productivity, software quality, cycle time, project costs and costumer satisfaction.

At process level, the entities to evaluate are the different phases of the development processes and the attributes of these phases which include cost, time, etc. In a measurement program, the organization has to identify the areas of measurement. The measurement objectives should be clear and well defined. Since project management is an important discipline in the proposed methodology, the key indicators must align with those used by the project manager to analyze and evaluate a project.

However, SPI key indicators may change and evolve. Over time, process changes can impact the way measurements are defined, the way measurements are collected, or the frequency of measurement collection and analysis activities. To facilitate this evolution and ensure that the measurements and indicators continue to provide meaningful information to managers, the continuous recording of project background information is important to: (1) facilitate the analysis and interpretation of measurements over time; (2) to establish links between measurement data sets over time and (3) to understand exactly how the measurements are evolving.

6. CONCLUSIONS

This paper describes ProPAM, a methodology for process improvement based on process and project alignment and supporting meta-models (PIT-ProcessM and PIT-ProjectM). ProPAM meta-models propose to bridge the gap between process description and development projects. PIT-ProcessM is used to specify the process life-cycle without forgetting that processes and projects are human-centered systems. The communication and collaboration actions are described in the meta-model, at an abstract level, through the role and activities relationship.

Agile and traditional processes are structured in phases and iterations, but actual process modeling approaches don't address this feature that is especially important in agile processes. Agile processes normally just plan iteration by iteration. Further, in approaches like BORE [13], life-cycle descriptions are most often treated as linear sequences, where crucial attributes of the process such as phases and iteration are not represented. PIT-ProjectM is used to plan the project as an instance of the process specified through the PIT-ProcessM with a life-cycle description based on phases and iterations. However, since every project has unique features and requirements, actors must have liberty and creativity to change the project. 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 our methodology, process versions are represented in a process versioning meta-model as a tree structure that supports mechanisms to allow an evaluation of the process changes along their several versions. The main goal is not only to get a description of the process really performed but, also important, is to analyze the effects on the organization of the improved process.

ProPAM was applied in a case study to help the organization improve its software product development process. The case study revealed that SPI isn't trivial, some improvements need organizations decisions, additional effort and time consuming activities. Sometimes, only after one year the effects of the proposed improvements will be observed. When passed projects data is needed to tune projects activities time and costs, it's important to continuously repeat the proposed improvement in future projects.

Further effort is required for better formalizing the methodology. Moreover, the methodology requires the support of a development environment, as much information has to be gathered and analyzed. Methodology concepts are being integrated in a project management tool (ProjectIT-Enterprise) of the ProjectIT research project. Future work will continue efforts to gather empirical data on its use will help refine the methodology and learn more about how to support the development process.

7. REFERENCES

1. Salo, O. (2005). Improving Software Development Practices in an Agile Fashion. *Agile Newsletter* 2/2005, Agile-ITEA, pp. 8.
2. Krasner H. (1997). Accumulating the body of evidence for the payoff of software process improvement-1997. Krasner Consulting. Retrieved on

September 8, 2006, from http://www.utexas.edu/coe/sqi/archive/krasner/spi.pdf
3. Software Engineering Institute. (1993). Capability Maturity Model for Software (CMM)", Version 1.1. Carnegie Mellon University.
4. Software Engineering Institute. (2002). Capability Maturity Model® Integration (CMMI[sm]), Version 1.1. Carnegie Mellon Software Engineering Institute.
5. SPICE. (1998). ISO/IEC Software Process Assessment Part 2: A model for process management, Version 1.0. SPICE Project.
6. Kuvaja, P., Simila, J., Krzanik, L., Bicego, A., Koch, G. & Saukkonen, S. (1994). Software Process Assessment and Improvement: The BOOTSTRAP Approach. Blackwell Publishers.
7. Beck, K. (2001). Manifesto for Agile Software Development. Retrieved on April 23, 2006, from http://www.agilemanifesto.org/principles.html
8. Gruhn, V. & Urbainczyk J. (1998). Software Process Modeling and Enactment: An Experience Report related to Problem Tracking in an Industrial Project. Proceedings of the 20th international conference on Software Engineering, 13-21.
9. Curtis, B., Kellner, M.I., & Over, J. (1992). Process Modelling. *Communications on the ACM*, 35, 75-90.
10. Krasner, H., Tirrel, J., Linehan, A., Arnold, P. & Ett, W.H. (1992). Lessons learned from a software process modeling system. Communications of ACM, vol. 35, n.9, 91-100.
11. Bolcer, G. & Taylor, R. (1996). Endeavors: A Process System Integration Infrastructure. International Conference on Software Process (ICSP4), Brighton, U.K.
12. Basili, V., Caldiera, G. & Cantone, G. (1992). A Reference Architecture for the Component Factory. ACM Transactions on Software Engineering and Methodology, 1 (1), 53-80.
13. Henninger, S. & Schlabach, J. (2001). A Tool for Managing Software Development Knowledge. Third International Conference on Product Focused Software Process Improvement, 182-195.
14. Heller, M., Schleicher, A., & Westfechtel, B. (2003). A Management System for Evolving Development Processes. Proceedings 7th International Conference on Integrated Design and Process Technology (IDPT 2003).