Process Management Methodology

Paula Ventura Martins
INESC-ID, CSI/Universidade do Algarve, Portugal

Alberto Rodrigues da Silva
INESC-ID/Instituto Superior Técnico, Portugal

INTRODUCTION

As organizations (involved or not in virtual organizations [VOs]) try to define their processes, they also recognize the need for continuous software process improvement (SPI) in order to improve the quality and productivity of products and to keep up their competitiveness. Although efforts have been made, organizations continue without success in achieving the intended goals. Recognizing that the most critical problems occurs during project activities, we strongly believe that both development process and project alignment can be the best practice to get better project results and improve organizations' development processes.

The authors assume the reader has a basic understanding of development processes (from now on, referred to as "process") and project concepts in general. Definitions and discussions about process and project are contained in (Krasner, 1992). "Process" in the most general sense, defines how an identified set of activities is to be performed in the context of the goals, objectives and constraints of an organization; "project" is defined as an instance of a process.

Nowadays, various development process models exist for efficient software development. They range from traditional, such as rational unified process (Kruchten, 2003) to agile processes, like eXtreme Programming (Beck, 2004). Traditional processes are much more rigid and prescriptive. *Agile processes* are a new and growing approach to software development methods. They attempt to offer an answer to the community asking for "lighter weight" along with faster software development processes. The focal aspects in agile processes are simplicity and speed. Development groups concentrate only on the immediately requested functions, delivering them fast, getting feedback and reacting quickly to business and technology innovations (Beck, 2004).

There is a great deal of literature about models for process improvement within organizations (e.g., CMM (SEI, 1993), CMMI (SEI, 2002), ISO/IEC 15504 (ISO/IEC 15504-7, 1998), Bootstrap (Kuvaja, 1994)). However, these models are very descriptive explaining essential attributes that would be expected to characterize an organization at a particular maturity level. However, they do not tell though how to improve and which are the specific means to get into a particular maturity level. This is the main reason for limited success in many SPI programs. Some studies recognize the need of further research on implementing SPI (El Emam et al., 1999).

The aim of this article is to propose a new methodology in the SPI domain. Target readers are people interested in process management and SPI. This subject concerns standalone organizations as well as collaborative organizations, such as virtual organizations. This article is structured in five parts. Part two presents a research review about process modeling and software process improvement. Part three describes the proposed methodology to support SPI based on process and project alignment. Part four presents an overview about SPI within the domain of VOs. Part five concludes and introduces future trends.

BACKGROUND

Research on software development process intends to define the process elements that constitute good practices, leaving implementation and enactment of the process to organizations. Also important is the way processes evolve within the changing needs of the development organizations. Currently, there is a lack of support on how SPI approaches address the problem about how processes are effectively applied and improved using knowledge from projects. Another challenge is how to control and validate important project changes that must be integrated within the project. So, the main challenges are: process representation, project representation and process and project alignment.

The quality improvement paradigm (QIP), the IDEAL model and ISO 155504 Part 7 methods propose an approach to managing an SPI initiative. QIP can be seen as a more detailed model drawing upon deming cycle applied in the context of software engineering (Basili, 1994). While QIP is an open approach for managing improvement, IDEAL improvement model is based on the process assessment results of CMM giving guidance on how to execute and manage an improvement program (McFeeley, 1996). In the ISO 15504 Part 7 model, the focus of SPI is expressed as: "software process improvement is based on process assessment results and process effectiveness measures" (ISO/IEC 15504-7, 1998, pp. 2).

Process modeling techniques are useful in defining the process, especially in the upper levels of maturity models like CMMI. Curtis (1992) discussed some approaches using process modeling to support process improvement, software project management and process-centered software engineering environments (PCSEEs). 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 (Krasner, 1992). In the domain of change management, the problem tracking system (PTS) is used to track errors and manage change request for the Wohnungswirtschaftliche Information System (WIS), a system built in a process-oriented way to support all business processes from the area of house constructing and administration (Gruhn, 1998). The Endeavors system is a flexible environment that allows users to create and evolve processes while a project is in progress (Bolcer, 1996). Although Endeavors supports most of the features in process definition languages and process modification, some problems arise about process coordination which can lead to chaotic and disorganized development processes. The BORE tool and methodology extends the experience factory concept (Basili, 1994) through rule-based process tailoring, support for process modeling and enactment as well as case-based organizational learning facilities. AHEAD system also supports the management and modeling of development processes. Additionally, AHEAD provides an integrated set of tools for evolving both process definitions and projects (Heller, 2003). In AHEAD, process evolved in terms of packages, which serve as units of version.

In Krasner (1991), the authors concluded that a flexible as well as active, intelligent, adaptive and orchestrated groupware that manages concurrent access to shared work spaces is a desirable goal for future process management systems. Collaborative environments are important for effective SPI but workshops are essential to a faster dissemination of process practices. Traditional SPI methods and approaches are based on final projects retrospectives. Since this work is performed in finished projects, project improvement can only be applied to future projects. There is a long-time span between problem identification and validation of the new process. Improvement opportunities resulting from projects must be analyzed, controlled and validated prior to dissemination in the organization practices. In agile principles (Beck, 2004), the project has reflections meetings in regular intervals. Cockburn proposes a reflection workshop technique (Cockburn, 2004), Dingsøyr and Hanssen have a workshop technique called postmortem review (Dingsøyr, 2002), whereas Salo and Abrahamsson discuss a Post Iteration Workshop (PIW) method (Salo, 2004).

Our work 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 (ProPAM), allows for a general vision on the current state of an organization development process, as well project alignment with the development process.

MAIN FOCUS OF THE ARTICLE: PROCESS AND PROJECT MANAGEMENT

The proposed methodology supports SPI based on process and project management alignment. Process and project alignment is defined as the degree to which project goals and plans support and are supported by the process practices. Moreover, it involves a real match between process practices and project activities, work 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 activities are executed. Furthermore, a modification may regard not only the considered activity, work product or actor, but it could also affect other elements having a dependence relation with the modified one. So, keeping track on

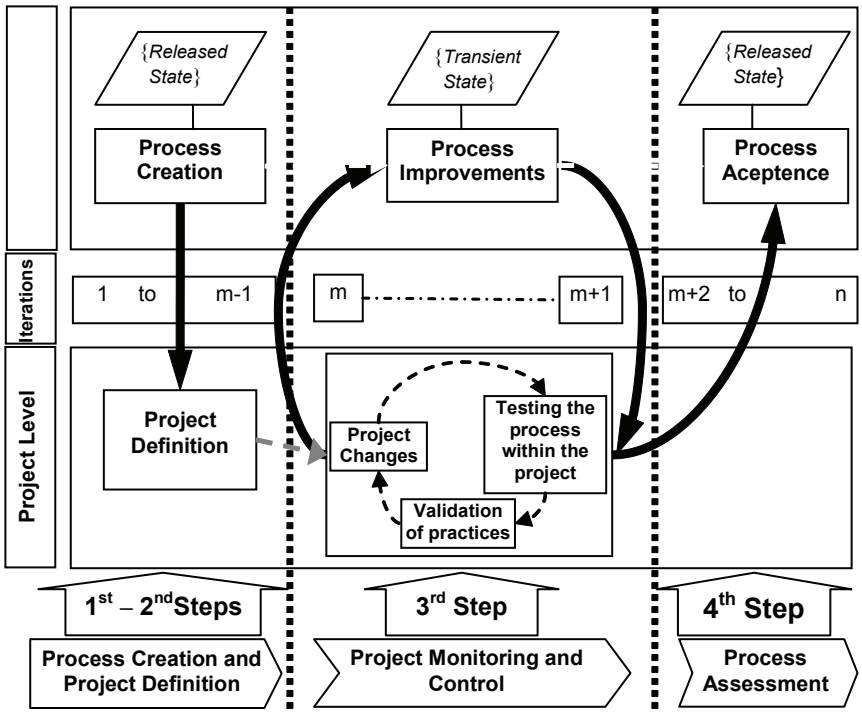
project execution, it is possible to detect changes in the project proposing improvements to the process considering that updates.

ProPAM is based on a modeling approach, since process and project modeling should be supported on a very high level of abstraction. The proposed architecture identifies and interrelate the necessary concepts to provide SPI based on process and project issues. The methodology is based on the four layered modeling architecture as defined by the Object Management Group (OMG, 2005). As shown in Figure 1, the ProPAM involves the following four steps:

- **Process creation:** The organization first creates a process using the ProjectIT Process Meta-model (*PIT-ProcessM*) (Martins, 2005) to identify different practices involved in the organizational process.
- **Project definition:** After the process is defined, it is used to create a project (process instance). This creates the initial project plan as defined in the process. The project manager then adapts this plan to the specific and unique project requirements and schedule project activities (*ActivityProject*) to actors (*Person*) with the right privileges (*RoleProject*) to perform them. Project and process alignment is carried out in this step and a ProjectIT Project Meta-model (*PIT-ProjectM*) was defined to support this dependency (Martins, 2005).
- **Project monitoring and control:** Team members execute the activities in their work list (*ActivityIterationProject*) and document their progress and changes to those activities. While enacting the project, *ActivityProjects*, *WorkProductProjects* and others elements are created and modified by team members, further improving the development process. This is often done through reviews proposed by process groups in workshops.
- **Process assessment:** The goal during this step is to demonstrate process suitability. It includes any process improvement required to correct any failings that are discovered.

PIT-ProcessM provides a set of generic concepts to describe any process. The meta-model consists of two complementary views: static and dynamic views. The static view represents formal process concepts like disciplines, work products, activities and roles. 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. It also allows decomposition of process life cycle into

Figure 1. Process and project alignment methodology



phases and iterations. This meta-model allows the specification of processes. A project is instantiated from a process, where a process represents reusable process practices at an abstract level. The process represents the best practices in software development but has no information about timing and resource allocation. For this reason, the project must specify exactly *who* must do *what* and *when*. The problem is that in real-world projects, multiple projects share the same process and are differentiated based on their specific elements, for example, actors, schedule, deadlines, resources, and so forth. Considering these important differences in a project management perspective, our approach presents a specific project meta-model—*PIT-ProjectM*. In this article, we just show the application of the meta-models. A more detailed description of the meta-models is presented in Martins (2005).

As proposed by agile processes, SPI is an iterative initiative during project time. 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 processes 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," three states are distinguished: transient, released and obsolete. The third step of the methodology, project coordination and monitoring, includes an SPI method performed during two iterations: (1) detect new improvements and create a new process version; (2) test and validate the temporary process version in the next iteration of the project and in case of success, continue playing it in future iterations. This method is applied throughout all iterations of the project, where the improvements follow a pattern performed across two iterations (Figure 1).

In the first iteration, the project team must perform their daily work and detect situations that lead to new practices in the project lifecycle. The data being collected include positive and negative aspects found by project team. The project manager has an important contribution in controlling the changes. At the end of this iteration, all candidate changes for process improvement are analyzed. If necessary a workshop is performed to get more knowledge and to 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 the 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 improvements may have been detected. The new version will be updated and the evaluation work performed in the second iteration is repeated.

In the end of the project, process improvements must be analyzed in a reflection meeting. The main goal is to analyze all improvement opportunities identified in the project and validate all the SPI actions accepted in workshops.

FUTURE TRENDS: SPI IN THE DOMAIN OF VIRTUAL ORGANIZATIONS

The concept of virtual enterprises aims to facilitate the collaboration of different companies to deliver a business opportunity. A number of traditional approaches in areas such as project management and SPI are proposed as an attempt to identify existing methods that can be used or adapted to tackle the problem of integrating companies wishing to be members of a virtual enterprise.

Latest tendencies in software development are the creation of virtual organizations, in which partners may be sought separately for each project. 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. This 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.

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 ProPAM methodology can be used in VOs environments but under some transformations in the following discussed subjects: (1) virtual process definition; (2) virtual project creation and (3) groups involved.

In the context of VOs, the *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.

Each organization defines their projects according to 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 for 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 changes can improve the virtual process, some members of participant organizations must perform the roles responsible by the SPI project.

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 ProPAM 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.

CONCLUSION

New methods and approaches for managing software projects and process are emerging. Models like CMM, CMMI and ISO/IEC 15504 are time consuming. They just describe essential attributes that would be expected to characterize an organization, but they do not tell how to improve. These models are theoretical, statistical, broad and manually described. Future SPI methods should be based in projects and organizational practices. Simplicity is a keyword in future trends since new approaches must be intuitive, skill-oriented and focused on projects. Daily work must be based on tools and templates. Software process must be efficient, without overhead and taking advantage from new technologies such as telecommunications and virtual work.

In this article, we propose the ProPAM architecture and principles. The approach presents an innovative proposal but also includes ideas of other research initiatives in agile processes. We believe that this methodology contributes to a real process and project alignment. We present two meta-models that can support that alignment. Currently, standard process meta-model, for example, SPEM, is not suitable for SPI, since its main goal is on process specification without any consideration for project management or analysis. The methodology concepts are being integrated in a project management tool (ProjectIT-Enterprise) of a research project (ProjectIT) from INESC-ID information systems group. In a VO domain, we discuss the extension of these meta-models to allow interactions between the virtual project and the locally executed projects of participating organizations.

REFERENCES

Basili, V., Caldiera, G., & Rombach, H. (1994). Experience factory. *Encyclopedia of Software Engineering*, Vol. 1 (pp. 469-476). John Wiley & Sons.

Beck, K. (2004). *Extreme programming explained: Embrace change.* Addison-Wesley Longman Publishing Co., Inc.

Bolcer, G., & Taylor, R. (1996). Endeavors: A process system integration infrastructure. *International Conference on Software Process*, Brighton, U.K.

Cockburn, A. (2004). *Crystal clear: A human powered methodology for small teams.* Addison-Wesley.

Curtis, B., Kellner, M., & Over, J. (1992). Process modeling. *Communications on the ACM*, *35,* 75-90.

Dingsøyr, T., & Hanssen, G. (2002). Extending agile methods: Postmortem reviews as extended feedback. *4th International Workshop on Learning Software Organizations,* 4-12.

El Emam, K., Fusaro, P., & Smith, B. (1999). Success factors and barriers for software process improvement. In R. Messnarz & C. Tully (Eds.), *Better software practice for business benefit: Principles and experience* (pp. 355-371). IEEE Computer Society.

Gruhn, V., & Urbainczyk J. (1998). Software process modeling and enactment: An experience report related to problem tracking in an industrial project. In *Proceedings of the 20th international conference on Software Engineering,* 13-21.

Heller, M., Schleicher, A., & Westfechtel, B. (2003). A management system for evolving development processes. In *Proceedings of the 7th International Conference on Integrated Design and Process Technology*.

ISO/IEC 15504-7. (1998). *ISO/IEC 15504-7 Software Process Assessment—Part 7: Guide for use in process improvement*. ISO/IEC TR 15504-7.

Krasner, H., McInroy, J., & Walz, D. (1991). Groupware research and technology issues with application to software process management. *IEEE Transactions on Systems, Man and Cybernetics*, *21*(4), 704-712.

Krasner, H., Tirrel, J., Linehan, A., Arnold, P., & Ett, W. (1992). Lessons learned from a software process modeling system. *Communications of ACM*, *35*(9), 91-100.

Kruchten P. (2003). The rational unified process: An introduction, 3rd Ed. Addison-Wesley Longman Publishing Co., Inc.

Kuvaja, P., Simila, J., Krzanik, L., Bicego, A., Koch, G., & Saukkonen, S. (1994). Software process assessment and improvement: The BOOTSTRAP approach. Blackwell Publishers.

Martins, P., & Silva, A. (2005). PIT-P2M: ProjectIT process and project meta-model. In *Proceedings of the OTM Workshop: MIOS+INTEROP 2005, Lecture Notes in Computer Science, 3762* (pp. 516-525). Springer.

McFeeley, B. (1996). *IDEALSM: A user's guide for software process improvement*. CMU/SEI-96-HB-001. Pittsburgh, PA, Software Engineering Institute.

OMG. (2005). *Software process engineering metamodel specification, Version 1.1*.

Salo, O. (2004). Improving software process in agile software development projects: Results from two XP case studies. *30th EUROMICRO Conference* (pp. 310-317). Euromicro.

SEI. (1993). *Capability Maturity Model for Software (CMM)*, Version 1.1. Pittsburgh, PA, Carnegie Mellon University.

SEI. (2002).*Capability Maturity Model® Integration (CMMI℠)*, Version 1.1. Pittsburgh, PA, Carnegie Mellon Software Engineering Institute.

KEY TERMS

Interoperability: The ability of two or more organizations or systems to be engaged in the process of ensuring that the organizations or systems exchange information and re-use the information, internally or externally.

Meta-Model: Defines a language for describing a specific domain of interest. A process meta-model provides a set of generic concepts to describe any process, defined in the next level (M1) of the layered architecture (OMG, 2005).

Performance Evaluation: An evaluation that compares actual project performance with that planned. The evaluation can be performed in terms of different criteria like people effort, resource use and production. It is used to redirect and improve project efforts and resources.

Process: A set of partially ordered activities performed in the context of the goals, objectives and constraints of an organization. Activities should be planned as part of a defined process, assigned to a role, allocated resources and produced/consumed work products.

Process Management: Organizational decisions in a development process in order to make changes and improve the process. To achieve these goals, pertinent techniques and tools are applied to a process to implement and improve process effectiveness, hold the gains and ensure process integrity in fulfilling customer requirements.

Project Management: A discipline that requires the application of knowledge, skills, tools and techniques during project activities to achieve the goals of a particular project, while optimizing the use of resources (time, costs, people, materials, etc).

Software Process Improvement: Continuous and iterative methodology in order to improve both software process and products through gained knowledge in project experiences. The main goal is to capture the activities, methods, practices and transformations that people use to develop software.