

ProPAMet: A Metric for Process and Project Alignment

Paula Ventura Martins¹, Alberto Rodrigues da Silva²

¹ INESC-ID, FCT/Universidade do Algarve
Campus de Gambelas, Faro, Portugal
pventura@ualg.pt

² INESC-ID /Instituto Superior Técnico
Rua Alves Redol, n° 9 –1000-029 Lisboa, Portugal
alberto.silva@acm.org

Abstract. Software Process Improvement (SPI) is one of the main software development challenges. Unfortunately, process descriptions generally do not correspond to the processes actually performed during software development projects. Process and project alignment is essential to really find out how process improvement is important to achieve an organization's strategic objectives. Considering this approach, this paper presents a new software SPI methodology designated by Process and Project Alignment Methodology (ProPAM). As a complement to be aware about project changes and facilitate the migration to an improved process, we propose a metric called ProPAMet to analyze the alignment between process and projects. To conclude, a case study contributed to validate the effectiveness of ProPAM and ProPAMet.

1 Introduction

Software process improvement (SPI) is a challenge to organizations trying 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].

There is a vast literature about process improvement approaches, such as: CMM [3], CMMI [4], ISO/IEC 15504 [5-7], BOOTSTRAP[8]. However, they don't tell though how to improve and which are the specific means to get into a particular maturity level. These approaches don't provide methods for process elicitation and modelling in order that projects follow specific development processes. They don't show how project practices and knowledge is gathered to contribute for process improvement. They don't explain the mechanisms of team members' collaboration to cope with changing contexts or react to existing problems. These are the main reasons for limited success in many SPI programs. Also important is the fact that some studies recognize the need of further research on implementing SPI [9].

Evaluation and, as a consequent, improvement of software processes would be impossible without software measurement [10]. Software process assessment is a

mean for organizations to identify their strengths, weakness, existing improvement activities and key disciplines for improvement. Measurement-based SPI enables organizations to determine the current state of their software process and to evaluate results of developed SPI programs. It also allows: (1) determining the effectiveness of applied processes [11]; (2) studying the effects of new practices introduced through improvement programs [12]; and (3) finally specifying process models that correspond to the processes actually performed [11].

All these factors allowed identifying several problems associated to existing SPI approaches, such as: (1) improvement actions focused on SPI models and ignoring organizational culture; (2) existing SPI models require several investments, such as: budget, time and human resources; (3) absence of key practitioners involvement result in resistance to change; (4) process descriptions generally do not correspond to the processes actually performed during software development projects; (5) existing SPI models don't provide methods for process and project representation; (6) existing SPI models identify *what* to improve but don't give any information about *how* to do it; and (7) no indicator about how project practices are diverging from the base process.

One of the contributions of this paper is to present a new SPI approach, **Process and Project Alignment Methodology (ProPAM)** is a SPI approach based on process and project data in order to detect misalignments between projects and supporting processes. The development of a metric for evaluating the accuracy of process and project alignment and the need to improve the process is another contribution that we intend to introduce in the domain of SPI. **Process and Project Alignment Metric (ProPAMet)** allows determining the alignment between processes and projects considered as an indicator to perform changes in base processes.

This paper is organized in the following sections. Section 2 describes briefly the proposed ProPAM methodology. Section 3 presents the proposed metric to analyse process and project alignment. Section 4 presents a case study performed in a Portuguese organization. Finally, Section 5 concludes and introduces future trends.

2 ProPAM Methodology

As mentioned in previous section, existing SPI models are insufficient to guide change in a constantly changing, constrained and increasingly unpredictable environment. Process and Project Alignment Methodology (ProPAM) directs attention to organization's needs for communication, coordination and collaboration within and between project teams. The methodology is about how the process and project are represented and how project teams acquire and use knowledge to improve work. ProPAM methodology proposes solving the problems faced in software development projects carried within the organizations. A critical feature in ProPAM is the integration of SPI activities with software development activities. This way, we considered project teams and projects as the baseline for improvement. A detailed specification of ProPAM can be found in [13].

As Figure 1 illustrates, ProPAM methodology includes SPI activities that intends to develop and implement the software process of an organization (process level). Nevertheless, SPI activities also include monitoring and tracking of software projects

(project level). At project level, the methodology proposes to assist organizations in its efforts to assess and manage problematic situations of specific projects, and develop and implement solutions that help manage these problems. The project level covers project(s) information needed to systematically support or reject many of decisions about the process. At process level, project’s feedbacks conduct to process reviews and iterative process improvement. The dynamic interplay between these two levels (project level and process level) show the synergy between the activities performed by project roles (project manager and team member) and the activities performed by the process roles (process manager) involved in SPI.

Figure 1 overviews the ProPAM methodology that includes the alignment between the process and project(s) illustrated through process and projects levels. The scope of the levels is well defined in order for process and projects actors collaborate on SPI programs. However, to manage the inherent complexity of these levels, namely ProPAM represented at process level, it is current practice to divide such models into views. In general, a view is defined as a projection of a process model that focuses on selected features of the process [14]. ProPAM is organized in two correlated and complementary views, the static view and the dynamic view that represent the behaviour at that particular level. Whereas the static view describes aspects of the methodology as core and supporting disciplines in terms of activities, work products and roles, the dynamic view shows the lifecycle aspects of ProPAM expressed in terms of stages and milestones.

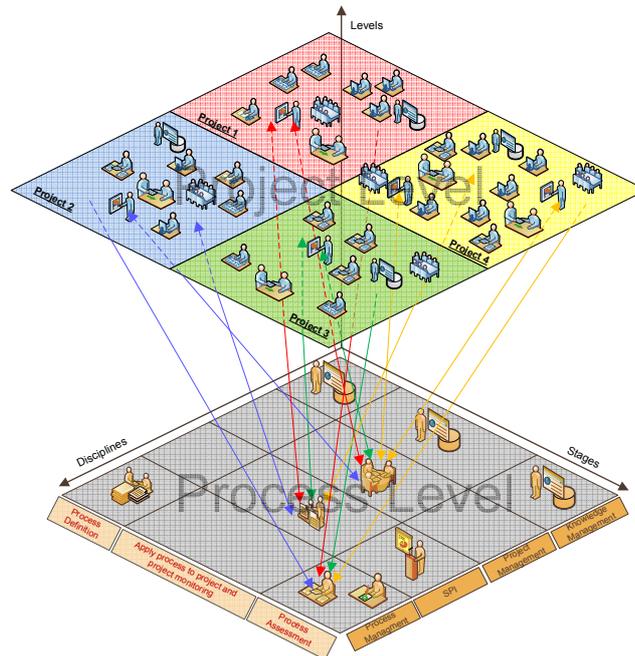


Fig. 1. Process and Project Alignment Methodology (ProPAM)

ProPAM static view integrates project management, process management, SPI and knowledge management (KM) disciplines, as illustrated in diagram of Figure 1. These disciplines assure alignment of projects with organization vision and goals, and the adopted and improved software process.

ProPAM dynamic view covers iterative process improvement through a SPI life cycle with three stages: (1) process definition; (2) apply process to project(s) and monitoring; and (3) process assessment and refinement. The **process definition** stage main goal is dedicated to an initial process specification through the application of the PIT-ProcessM metamodel specified in Figure 2. **Apply process to project(s) and monitoring** stage involves planning and executing the project within the base process best practices. It also provides assurance that the project is progressing according to the base process or reveals the need to take SPI actions because the activities performed by team members are different from those specified in the process. Project problems may occur and a new set of practices must be imposed or the process manager detects that new practices are needed. In the **process assessment and refinement** stage, initially, the project manager and process manager analyze project data and produce assessments focused on project issues and process issues, respectively. After that, the results gathered during assessments enable improvements and consistent refinements of the base process creating a new process version.

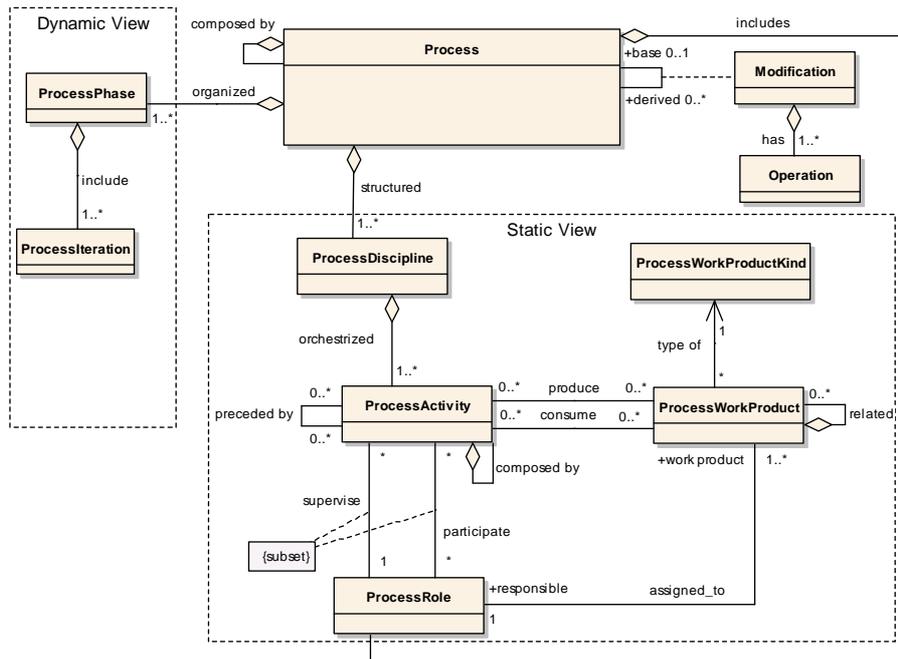


Fig. 2. ProjectIT Process Meta-model

3 Process and Project Alignment Metric (ProPAMet)

Process descriptions generally do not correspond to the processes actually supporting software development projects. They just represent high-level plans and so, do not contain the concrete information necessary for a software project. This lack of alignment between the process and project(s) results from processes 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.

However, progressive modifications in projects can cause misalignments with the original 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. ProPAM methodology provides mechanisms to detect misalignments between processes and projects through detection of changes and innovations in project's activities. This mechanism is based on a metric that allows defining the alignment degree between process and project.

Process and project alignment is defined as the degree to which the project activities support and are supported by the process practices. Moreover, it involves a real match between process practices and projects activities, products and actors. We propose a metric where the process is considered the reference, and the measure provides a balanced assessment of the fidelity of matches and gaps.

Project metrics goals are important to improve project-by-project performance, divisional/sector performance, or organizational performance. Process metrics are also important to quantify attributes of the development process and the development environment. However, none of these approaches allows identifying the similarities between the features considered in both domains (process and project). The alignment metric that we propose intends to characterize how closely the projects are related to their base process.

The **Process and Project Alignment Metric (ProPAMet)** evaluates the mapping features of one project to features of the process. The alignment measure intends to evaluate the correctly match between process and project features (aligned features) divided by the total number of features identified in a project. The measurement process contains the following phases:

- identification and classification of project features by categories, considering: (1) unaligned features in process entities and (2) aligned features in process entities;
- calculating an aligned features value;
- calculating general project features value;
- calculating final process and project alignment value.

For measurement purposes, project features are organized according to process entities in the following five categories: phase, discipline, role, work product and activity. The formula to derive process and project alignment can be written:

$$\text{ProPAMet}(AF, TPF) = AF/TPF . \quad (1)$$

Where:

AF = number of project's entities that have a correspondence in process' entities (Aligned Features).

TPF = number of project's features (aligned and unaligned features) in all five categories (Total Project Features).

In (1) each term **AF** and **TPF** is calculated by formulas (2) and (3). However, features from different categories have a different impact in the process. To consider the relevance of each category, project features must be multiplied by a constant weight w_i in formulas (2) and (3). Each category has its own constant weight w_i ($i=1,2,3,4,5$) as presented on Table 1.

Table 1. ProPAM weights

Category	Weight factor
Phase	0,2
Discipline	0,25
Role	0,1
Work Product	0,05
Activity	0,05

To derive the **AF** function, first identify all project entities that have correspondence in process entities. Then weight each aligned feature based on one of the five categories. The sum of these weights is called the aligned features function (AF):

$$AF(i,a,w) = \sum (a_i \cdot w_i) . \quad (2)$$

In (2) each term **i**, **a_i** and **w_i** represent:

i = features are classified through five categories (phase, discipline, role, work product and activity)

a_i = number of project aligned features classified in category **i**.

w_i = weight assigned to category **i**.

However, some features could have no correspondence in process entities for respective categories. Then, consider project features as the project activities with and without correspondence in process entities. Then, weight each project feature based on one of the five categories (Table 1). The sum of these weights is called the total project features (TPF):

$$TPF(i,p,w) = \sum (p_i \cdot w_i) . \quad (3)$$

In (3) each term **i**, **p_i** and **w_i** represent:

i = features are classified through five categories (phase, discipline, role, work product and activity)

p_i = number of project features (aligned and unaligned) classified in category **i**.

w_i = weight assigned to category **i**.

Over time, process and project misalignment happens when projects practices gradually changes to a point where differences to the base process are considered relevant. ProPAMet, a metric to evaluate process and project alignment enables to compare performed practices in current project with practices of the base process. The metric compares the base process at time t_0 with practices in projects at time t_i , allowing notifying practitioners about differences between process and project practices. When the ProPAMet threshold is crossed, differences are significant and it is a recommended to start a new SPI program which probably will conduct to an improved process version.

Figure 3 illustrates the second stage (apply process to projects and monitoring stage) of a hypothetic SPI program. Three iterations were executed until achieve organization goals. Iteration ends when the metric threshold is crossed and a new process version is delivered. This cycle concludes when the SPI program goals are fulfilled.

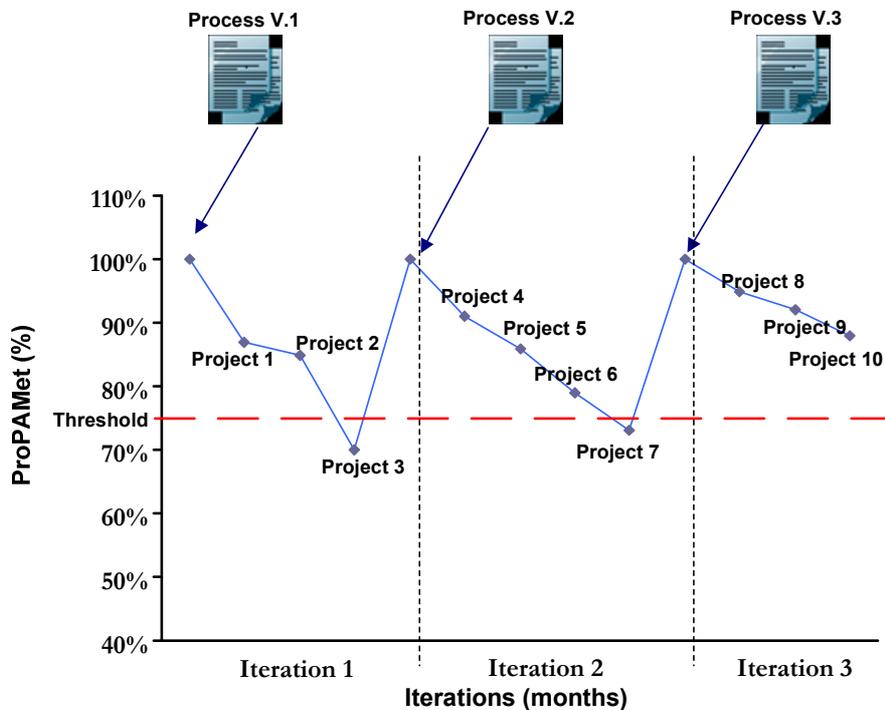


Fig. 3. Example of a SPI program with three iterations

4 Case Study

The purpose of the case study was to evaluate the effectiveness of ProPAM as a new methodology for SPI in small and medium organizations. We collaborate with a

Portuguese software house that had demonstrated interest to define and improve their software development process. The case study includes the observation of three different projects and the application of the proposed methodology – ProPAM – to define and improve their software development process.

A SPI program was conducted in order to control and analyse projects developed by this organization. The SPI program was organized in three stages. The first stage was dedicated to an initial process specification at process level. While in the second stage several activities had been realized at process and project level. At project level, three projects had been under inspection to detect, introduce and validate new software development practices. Then, these practices had been analysed at process level as candidates for future improvements in the base process. Final stage main is dedicated to specify the improved process and includes a final feedback meeting to discuss introduced practices.

SPI roles planned and performed improvement activities over a period of ten months, which resulted in the definition of the process (a process model, process documentation guideline) and a knowledge base (documents, guidelines, projects data, template library). At the end, the changed process had been presented to senior manager and project teams and further modified and improved based on their feedback.

Critical work of a SPI program was developed during the second stage of the SI program. At project level, the project PTF had been monitored during 12 iterations that lasted one, two or three week's time each. Project NGRID and PIS were organized in fewer iterations, respectively 5 and 4 as showed in Table 2. This table also provides a profile of attributes for the three projects.

Table 2. Main features of the three inspected projects

Characteristic	Project NGRID	Project PIS	Project PTF
Project name	NGRID	PIS	PTF
Application	Web-based application	Web-based application	Portal (front-end and back-office)
Duration	7 weeks (planned) 10 weeks (actual)	6 weeks (planned) 9 weeks (actual)	18 weeks (planned) 25 weeks (actual)
Number of Iterations	5 iterations	4 iterations	12 iterations
Iteration length	5 x 2 weeks	1 x 2 weeks 1 x 1 week 2 x 3 weeks	3 x 2 weeks 1 x 1 week 6 x 3 weeks
Project Team Size	5	4	4

At process level, only an iteration took place during the second stage. As we can see, at process level, iterations act in a different time scale expressed in months. In this case study, this iteration lasted six months. The nature of the project and process level iterations won't necessarily change much, so we recommend at least one SPI program each year. Figure 4 illustrates the difference between the time scale of the iterations at process and project level. It also identifies main activities and demonstrates the interaction between these two levels.

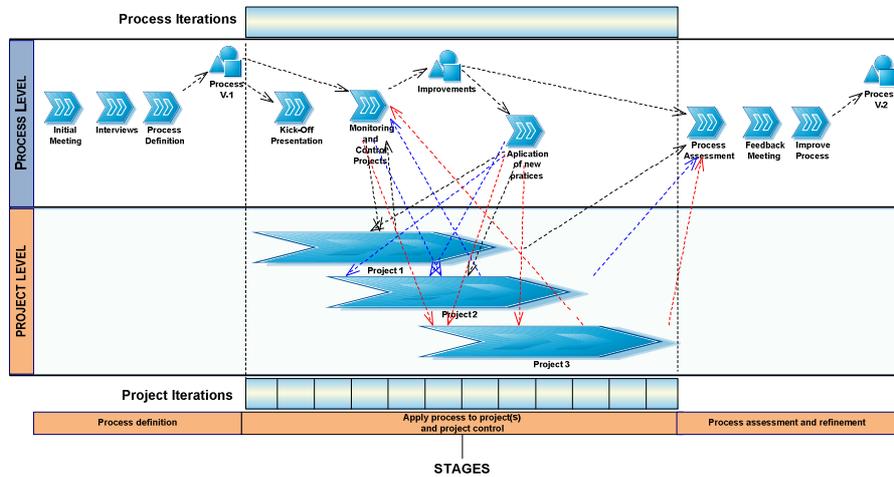


Fig. 4. SPI program performed at the Portuguese software house

During the period of the pilot case study from September 2006 to July 2007 we collect data from these three projects. All the data presented here was obtained through analysis of several projects’ work products and SPI documents. The data collected were analysed statistically, and proposals were developed for improving the software development process based on the results of an analysis of the qualitative data collected in the assessment and other quality improvement findings from developed projects.

Finally, after analysing data collected through the three projects and comment final metrics results, it is the moment to verify the impact of the proposed practices in the base process. In accordance with ProPAM methodology, ProPAMet is a metric which allow identifying the degree of alignment between projects and the correspondent process. The main objective is to determine the degree of alignment in order to advice improvements in the base process. A high degree of alignment indicates that projects practices are highly synchronized with correspondent process.

Here the challenge occurs under changing project practices, such as the ones introduced in these three projects (case study) that could result in a shift to a new process version. When such changes take place, ProPAMet is used to evaluate the degree of alignment or consensus between projects and the correspondent process. ProPAMet calculus involved identification and classification of the features from projects and from the base process according to specific categories (see table 1). To determine the match between process and projects, two functions were applied (equation 2 and 3) to determine aligned features (AF) and total project features (TPF) (see section 3). In total and relative to the base process, project NGRID included the following unaligned features: 1 discipline, 10 work products and 6 activities. Whereas the corresponding values were 1, 13 and 7 in project PIS and 1, 18 and 10 in project PTF (table 3). Projects PIS and PTF included one more role in the unaligned features, the Web Designer.

Table 3. Existing and new practices organized by disciplines

Analyse and design			
Interviews	+	+	+
Prototyping	+	-	-
Use cases	-	+	+
Requirements specification	+	+	+
Issue and change request management	+	+	+
Modelling	±	±	+
Design information	-	-	-
Requirements management	±	±	±
Requirements traceability through design	-	±	±
Development			
Write code	+	+	+
Test-Driven Development (TDD) techniques	-	-	-
Pair programming	-	-	-
Pair programming training	-	-	-
TDD training	-	-	-
Tests			
Unitary tests	±	±	±
System tests	±	±	±
Final client test	±	±	+
Pre-production debug	+	+	+
Test cases	±	±	±
Client participation on test cases	-	-	±
Independent tester	-	-	-
Peer review	-	-	-
Cross-reference between requirements and test cases	-	-	-
Defects management	±	±	±
Deployment			
Prepare client installation	+	+	+
Client installation	+	+	+
Prepare project presentation	+	+	+
Present project to client	+	+	+
Project Management			
Kick-off meeting	+	+	+
Prepare project meetings	-	-	-
Iteration meetings	+	+	+
Elaborate commercial proposal	±	±	±
Historical data	-	-	-
Estimation	-	-	-
Planning/replanning	±	±	±
Tracking project	-	-	±
Periodic reports	±	±	±
Risk management	-	-	±
Quality management	±	±	±
Software configuration management	-	-	-
	NGRID	PIS	PTF

Figure 5 shows the variance of the ProPAM metric (ProPAMet) within the three projects. The graphic illustrates a decrease in the value of ProPAMet from project NGRID to project PTF and, consequently, shows how changes proposals influenced project practices and conduced to process improvement.

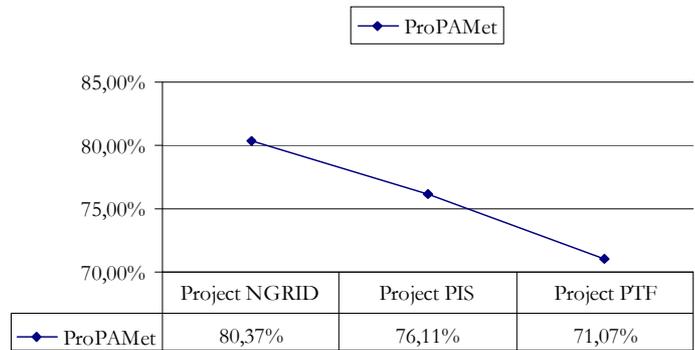


Fig. 5. ProPAM metric (ProPAMet) for each project

5 Conclusions

Currently, existing process metamodels are not suitable for SPI, since their main goal is on process specification without any consideration regarding project changes to improve the process. These problems redirected our efforts to define two metamodels (PIT-ProcessM and PIT-ProjectM) that are applied in process and project specifications and respective alignment. On the other hand, within ProPAM, PIT-metamodels contributed to the solution of previously identified SPI problems. So, these metamodels had been proposed to fulfil the following requirements: (1) provide support for process definition and improvement; (2) specify projects based on a previous process description; (3) track project issues back to the process. Round trip was also an important feature, since we used reverse engineering to improve the process model based on the changes introduced in the project description (process and project alignment).

The contribution of this thesis was not just a modelling approach to align process and project specifications, within ProPAM, we also proposed a mechanism to process evolution based on the changing needs of the software development organization. The case study, and consequent results evaluation, demonstrated the effectiveness of ProPAM to improve an organization software process.

Concerning ProPAMet metric, this paper introduced a metric that graphically helps to decide about process and project misalignments that will conduct to SPI actions to improve an organization process. When an organization applies ProPAMet metric during successive software projects, it obtains a feedback about how project practices are diverging from proposed process practices. Therefore, we highly recommend using ProPAMet during all developed software projects for two reasons: (1) evaluate if projects changes proposed by SPI programs are relevant to justify a new and

improved version of the base process and (2) determine if current project practices are not aligned with base project, justifying a new SPI program to improve the process.

References

- [1] O. Salo, "Improving Software Development Practices in an Agile Fashion," in *Agile Newsletter 2*, Agile-ITEA ed, 2005, pp. 8.
- [2] H. Krasner, "Accumulating the body of evidence for the payoff of software process improvement-1997," Krasner Consulting 1997.
- [3] SEI, "Capability Maturity Model for Software (CMM), Version 1.1," Carnegie Mellon University 1993.
- [4] SEI, "Capability Maturity Model Integration (CMMI), Version 1.2," Software Engineering Institute, USA 2002.
- [5] ISO/IEC, "15504-2 Information technology - Software process assessment – Part 2: A reference model for processes and process capability," July ISO/IEC TR 15504-2, 1998.
- [6] ISO/IEC, "15504-5 Information technology – Software process assessment – Part 5: An assessment model and indicator guidance," ISO/IEC JTC1 / SC7, 1998.
- [7] ISO/IEC, "15504-7 Information technology – Software process assessment – Part 7: Guide for use in process improvement," International Organization for Standardization ISO/IEC TR 15504-7, 1998.
- [8] P. Kuvaja, J. Simila, L. Krzanik, A. Bicego, G. Koch, and S. Saukkonen, *Software Process Assessment and Improvement: The BOOTSTRAP Approach*, Blackwell Publishers, 1994.
- [9] K. El Emam, P. Fusaro, and B. Smith, "Success Factors and Barriers for Software Process Improvement," *Better Software Practice For Business Benefit: Principles and Experience*, Publisher, City, 1999, pp. 355-371.
- [10] L. Arthur, *Improving Software Quality: An Insider's Guide to TQM*. New York, 1993.
- [11] S. Pfleeger and H. Rombach, "Measurement Based Process Improvement," *IEEE Software*, Publisher, City, 1994, pp. 8-11.
- [12] R. Van Solingen and E. Berghout, *The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development*, McGraw-Hill. Cambridge, UK, 1999.
- [13] P. V. Martins and A. R. Silva, "ProPAM: SPI based on Process and Project Alignment," presented at 2007 IRMA Internacional Conference, IGI Publishing, Vancouver, 2007.
- [14] M. Verlage, "Multi-view modeling of software processes," presented at Third European Workshop on Software Process Technology, Springer-Verlag, Grenoble, France, 1994.