A Case Study applying Process and Project Alignment Methodology

Paula Ventura Martins[1] **& Alberto Rodrigues da Silva**[2]

[1] INESC-ID, CSI / Universidade do Algarve
Campus de Gambelas
Phone: +351.289.800905
Zip 8005-139 Faro, PORTUGAL
pventura@ualg.pt

[2] INESC-ID / Instituto Superior Técnico
Rua Alves Redol, nº 9
Phone: +351.21.3100307
Zip 1000-029 Lisboa, PORTUGAL
alberto.silva@acm.org

Abstract

Software Process Improvement (SPI) is one of the main software development challenges. However, SPI standards and models (CMMI, SPICE) have not been always adopted with success. The current problem is a lack of strategy to implement successfully these standards and models. To undertake this objective is essential observe real life experiences and detect process and project mutual relationships. Without this alignment it will not be possible to find out how process management is really important to achieve organization's strategic objectives. This paper 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 the specification of an organization development process, as well process and project alignment. ProPAM presents the following life cycle: (1) process definition; (2) project definition considering a base process model; (3) project coordination and monitoring and (4) process improvement assessment. This paper also provides an overview of the action plan to be taken within the software organizations that intent to conduct a SPI initiative. This plan includes two distinct phases: (1) specify the development process and (2) analyze projects, starting an SPI effort. In order to evaluate ProPAM, a study case is undertaken. The case study is performed following the action plan and presents all the steps of the ProPAM. Final results show that, when the organization started using ProPAM, process and project alignment reduced project planning time and effort. ProPAM also introduced new organizational practices that result in a SPI program.

Keywords: Software Process Improvement, Project Management, Meta-models

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]. SPI involves a series of small steps, where each change has to be introduced in the habits of the team members. Although all groups involved in a SPI initiative are important, project team members are the most important because they actually execute the process (project) and they are the central sources of input for a SPI program [8]. Project team members have an important role in planning, implementing and adapting process activities through a project.

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 [9]. 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.

Most recent trends show that the software process community is ready to accept many process modeling languages and also agree that graphical process modeling would help to alleviate process modeling for non language experts. At same time, we also observe more emphasis on researches about SPI issues. Despite the great progress observed on software process research, process research has not had as much impact as expected on actual industrial software process enactment (also known as project) based on software process and further process improvement. Since software processes are human-centered requiring interaction and cooperation, process modeling must also focus on interacting behaviors among actors, including the variety, uncertainty and creativity of the actors involved. Considering these foundations, project must not be seen as a simple instance of a development process. Every project has unique characteristics and requirements that extend the knowledge process, process practices can only be used as a guide to create project plans. On the other hand, projects can help in ensuring process improvement once they are aligned with a base process.

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. Section 3 sketches architecture of the proposed framework to support process and project alignment in iterative (traditional and agile) SPI approaches. Section 4 presents the action plan presented to the enterprises to perform work join work. Section 5 describes our case study. 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. Some of these approaches include CMM [3], CMMI [4], SPICE [5] and Bootstrap [6]. However, these models are very descriptive in the sense that they explain essential attributes that would be expected to characterize an organization at a particular maturity level, but they don't specify neither how to improve nor the specific means to

get into a particular maturity level. But, as discussed by the research community, also important is the way processes evolve with the changing needs of the development organization. In addition, projects must adopt the process with some level of detail for the organization. Process modeling techniques are useful in defining the process, especially in the upper levels of maturity models like CMMI. Curtis, Kellner and Over discussed some approaches using process modeling to support process improvement, software project management and Process-centered software engineering environments (PCSEEs) [10].

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 [11]. 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 [9]. The Endeavors system is a flexible environment that allows users to create and evolve processes while a project is in progress [12]. 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 [13] through rule-based process tailoring, support for process modeling and enactment and case-based organizational learning facilities [14]. The AHEAD system also supports the management and modeling of development processes and provides an integrated set of tools for evolving both process definitions and projects [15]. In AHEAD, process evolved in terms of packages, which serve as units of version.

In [16] 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 software 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 [17]. Since this work is performed in finished projects, the any project improvement can only be applied in future projects. There is a long time span between the problem identification and the validation of the new process. The improvement opportunities resulting from projects must be analyzed, controlled and validated prior to being disseminated in the organization practices. In agile principles [7], the project has reflections meetings in regular intervals. Cockburn proposes a reflection workshop technique [18], Dingsøyr and Hanssen has a workshop technique called postmortem review [19], whereas Salo and Abrahamsson discuss a Post Iteration Workshop (PIW) method [20].

3. PROPAM

Our investigation 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 a general vision on the current state of an organization development process, as well project alignment with the development process.

Process and project alignment gives project team members more productivity, especially in project planning phase. Although process and project are different, they can be integrated as they are related and share some concepts [21]. By aligning the process and the project, the project will be performed based on a specific process. Additionally, keeping track on project execution, it is possible to detect changes in the project and propose improvements to the process considering these updates.

ProPAM is based in 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 concepts necessary to provide SPI based on process and project management issues. This section describes the components features essential to process and project alignment. Process and project alignment formalization has four components:

- Process modeling allows an easy way to graphically construct a process;
- The project modeling (based on a process) provides the necessary coordination facilities for process and project alignment;
- Project control and monitor allows observing changes in the project that are candidates to SPI. As complement, process versioning allows to create process versions based on the proposed improvement to process;
- Process assessment allows evaluating the benefits from process improvements.

The methodology is based on the four layered

architecture of modeling as defined by Object Management Group (OMG). In the level M2, Software Process Engineering Meta-model (SPEM) is an OMG standard for "process specification, without specific models or constraints for any specific area or discipline, such as project management or analysis" [21]. The actual processes enactment, that is, project planning and executing (level M0), isn't in the scope of the SPEM. Considering that our goal is an iterative SPI approach with an emphasis on process and project management, SPEM isn't the right meta-model approach to the proposed methodology. SPEM also address some concepts that are not essential in a project management perspective. A simplified meta-model with extension mechanism to integrate specific project management issues must be considered.

The next four subsections present the features of the four components of the ProPAM.

3.1. Process Definition Component

A 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 [21]. ProjectIT Process Meta-model (PIT-ProcessM) architecture defines the classes that correspond to elementary process concepts, allowing process creation or modification [22]. Two complementary views show those static and dynamic process elements. In the static view are represented the concepts related to process *Disciplines*, like *WorkProducts*, *Activities* and *Roles*. Dynamic view of the meta-model is about how a process lifecycle is organized, e.g., phases and iterations. Process patterns that are associated to different moments of a process should be considered in a temporal scale. Additionally, all the process elements should be associated to a particular moment of a process lifecycle. An interface between this views is made by the class *Activity_Iteration*, where are specified the *Activities* that belongs to an *Iteration*.

Process definitions are created as instances of PIT-ProcessM. Figure 1 presents the PIT-ProcessM architecture.

Static View. An *Activity* represents the work performed by a *Role*. But an *Activity* can be decomposed in small work units, also called *Activities* to unlimited deep of nested work. This concept is represented by the reflexive composed by aggregation. Control and data flow between *Activities* is defined by the reflexive preceded by association. *Activities* produce and consume *WorkProducts*, which can also be formed by a set of small *WorkProducts*. Each *WorkProduct* is identified by a *WorkProductKind*, e. g., a document, a model, a source code, and so on. *Activities* are organized according to a common "theme" in *Disciplines*.

Dynamic View. The dynamic view identifies how process can be managed in terms of *Phases* and *Iterations*. *Phases* are defined with the additional constraint of sequentiality, with a series of milestones spread over time and often assume minimal overlap of their activities in time. Each *Phase* include some *Iterations*, which are activities flows but with small goals.

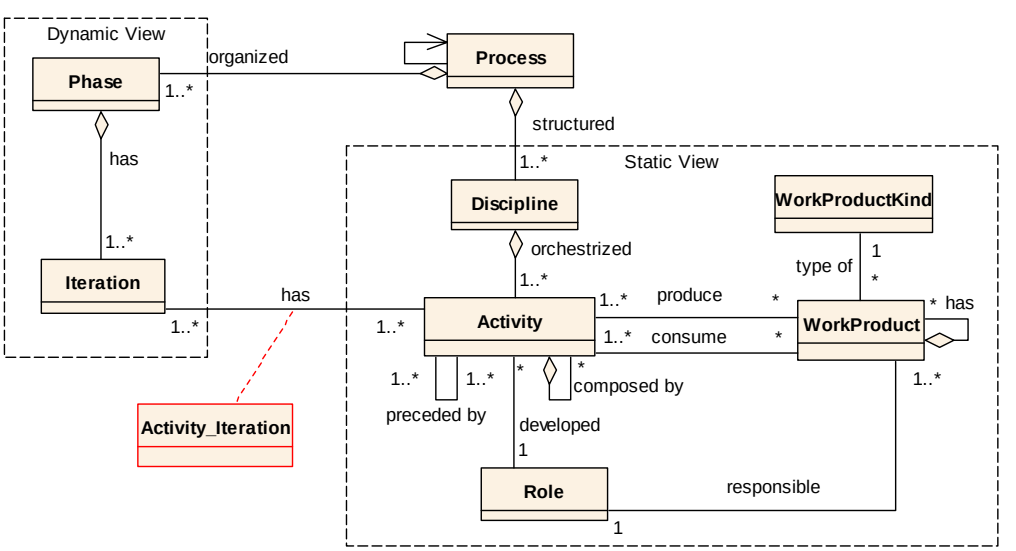

Figure 1: ProjectIT Process Meta-model (PIT-ProcessM)

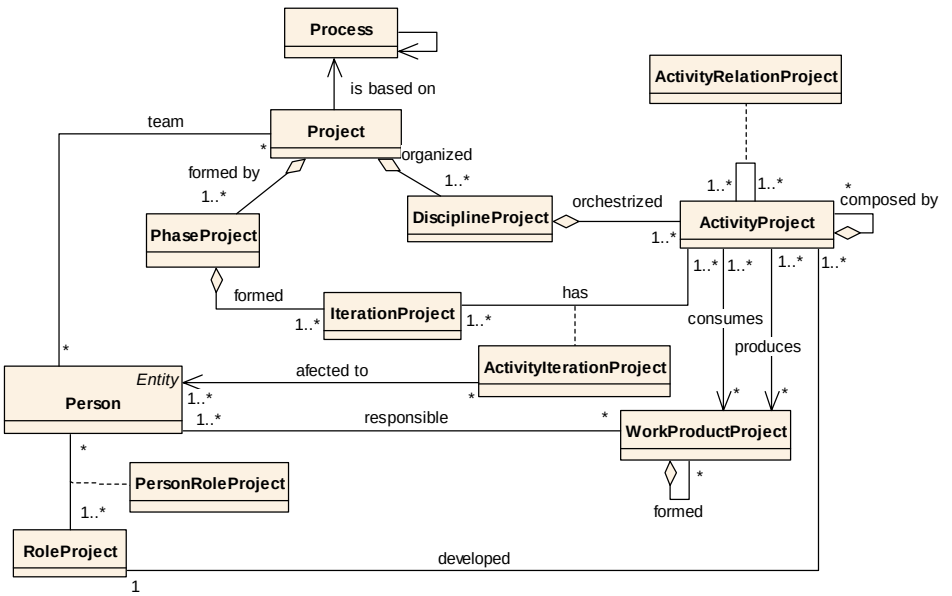

Figure 2: ProjectIT Project Model (PIT-ProjectM)

3.2. 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 (level M0 of the layered architecture), multiple projects share the same process and are differentiated based on their specific elements, e.g. persons and the resulting relationships. Considering this differences and process and project alignment carried out in this phase, our approach includes a ProjectIT Project Meta-model (PIT-ProjectM) to support this dependency [22].

A project specification is based on the initial process definition model. *Projects* are frequently divided into more manageable components or subprojects, although the individual subprojects can be referred by themselves as projects and managed as such. *Projects* have associated specific information on *Phases*, *Iterations*, *Activities*, *Roles* and *Workproducts*.

In a modeling perspective (figure 2) the differences between PIT-ProjectM and PIT-ProcessM met models are in the following classes: (1) *ActivityIterationProject*, which defines the *Activities* to perform in each *Iteration*; (2) *Person*, team members; (3) *PersonRoleProject* that defines the association between *Persons* and project *Roles*. Associations between the classes *Person*, *WorkProductProject* and A*ctivityIterationProject* allow each person to visualize its responsibilities (activities and products), and verify relations to other persons work.

Team members can manage their time, place priorities in their activities, made decisions supported by data. Project manager has a global vision of project performance, being able to observe details in *Activities*, *Iterations* and team performance.

3.3. PROJECT COORDINATION AND MONITORING COMPONENT

The third component 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 (*ActivityProject*, *WorkProductProject*, *RoleProject*, etc) are an instance of process elements (*Activity*, *WorkProduct*, *Role*, etc), project team members have the liberty to create *ActivityProject* and *WorkProductsProject* specific to a project. These changes are detected through the SPI actions performed by the process group, described later (subsection 3.3.2). Process improvement requires a process group capable of integrating new practices, adopted by project team members, into the current project. 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 component is composed by two different subcomponents. The first one

(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 subcomponent (Process Versioning) subsumes that changes are accepted and that it is necessary to keep historical information about several process versions.

3.3.1. PROJECT ITERATIONS

The proposed SPI method is developed in 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 applying it in future iterations. The SPI method is developed throughout all iterations of the projects, but the improvements follow a pattern that is performed across two iterations (figure 3). The groups involved in SPI consist of the software development team, project manager and the process group. If necessary, at the end of iterations, a workshop is realized to review the changes proposed by team members and also to process group notify about changes integrated in the process.

First Iteration: In the first iteration, the project team must perform their daily work and detect situations that conduce to new practices in the project lifecycle. The data collected includes 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 change candidates to improve the process are analyzed by the process group and if necessary a workshop is performed to get more knowledge and present the improved process (new process version).

Second Iteration: 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. In 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.

3.3.2. PROCESS VERSIONING

As proposed by agile methods, SPI must be an iterative initiative along 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 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 [23], three states are distinguished: *transient*, *released* and *obsolete*.

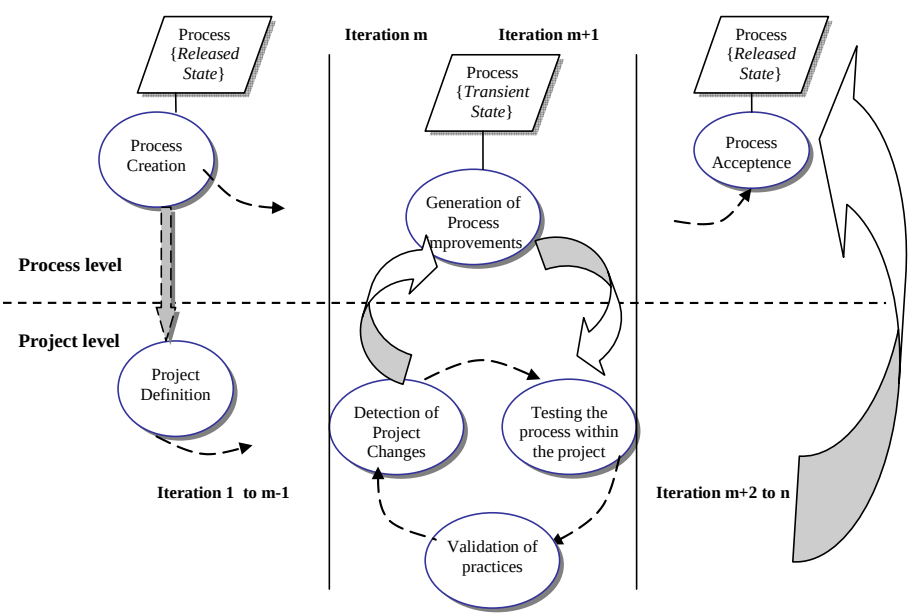

Figure 3: Process and project alignment

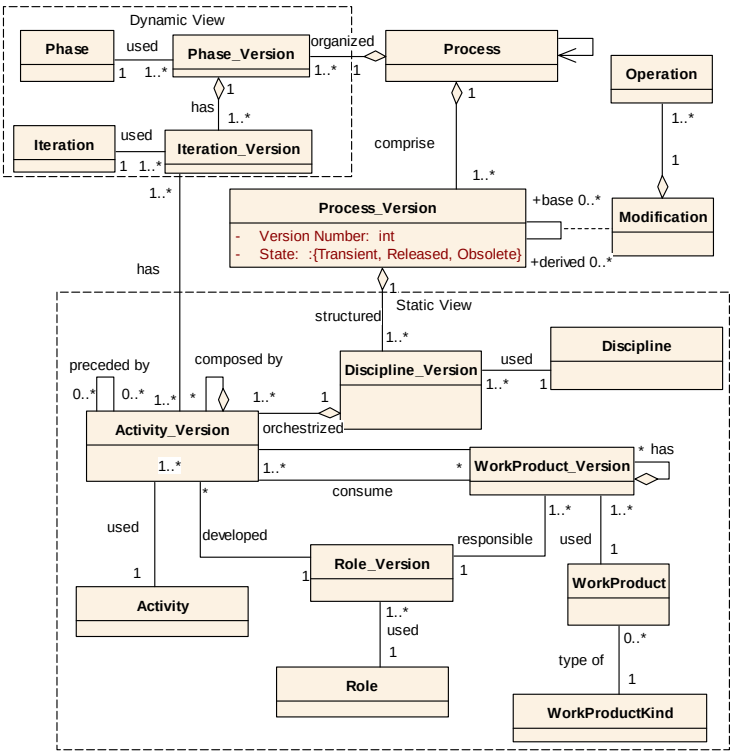

Figure 4: PIT-ProcessM Versioning

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* is not allowed: (1) to create descendants versions; (2) to create projects based on that version; (3) have more than one project based on that process version and (4) the version be referenced by other version. A process version in state transient can has only one instance if the process improvement results from a project change. The project must also pass to a different state (*transient* state), when the project is associated with a SPI evaluation. Therefore, projects can also have associated states: *normal* and *transient*.

Finally, when a version is accepted is 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 is state can be 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 no new project can be based on it. An obsolete version can be deleted if the version has no descendents, has no derived projects and isn't referenced by other versions. But first 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. Figure 4 presents the main constructs of the extended meta-model. A process class includes an 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 its versions. The PIT-ProcessM was updated to include the modifications operations applied to a process in creating a new version. In the meta-model, original elements from PIT-ProcessM like *Phase*, *Iteration*, *Discipline*, *Activity*, *Role* and *WorkProduct* are replaced by its versions classes. The associations represented in PIT-ProcessM between these elements are now performed at version level. This means that associations are between two versions elements. The elements are represented by an association with an element version, since each element is used in one or more process versions.

3.4. PROCESS IMPROVEMENT ASSESSMENT

In the final phase (process improvement assessment), progress is evaluated through 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 conducted successfully in many organizations. However, process managers ask themselves important questions following such activities, such as: (1) How to evaluate when a new process version meets the organization 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 ProPAM, 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 a future application of the software process.

One's prospects for success in executing and improving software process activities rise significantly when decisions can be based on quantitative information that can be obtained only 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. Some of the most used approaches to measuring SPI are: (1) the Goal-Question Metric [24]; (2) the Grassroots Effort [25]; (3) targeting for a maturity level and (4) team members perception. However, all these approaches present some limitations to effectively measure SPI. In the Goal-Question Metric, it's difficult to establish useful measurement programs from a business perspective. In the Grassroots Effort, project managers learned to establish a baseline through a series of dedicated measurements, starting with relatively simple indicators and giving priority to the practical use of data. An alternative approach is targeting for a maturity level, with this propose improvement goals become elusive, making it harder to mobilize the organization. Another option is to rely on team members' perception. This approach might maintain organizational commitment, but it provides little information to manage and improve the change process.

Prudently used, feedback approaches motivates SPI and supports the assessment of SPI strategies and tactics. 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. By monitoring changes in key indicators, process managers and process groups see the efficiency of the new process version compared to previous results of the original process adopted.

Once project management is an important discipline in the proposed methodology, the key indicators must be those used by the project manager to analyze 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.

However, these SPI keys 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 caption 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.

4. ENTERPRISES COLLABORATION ACTION PLAN

Many improvement programs fail because SPI success has more probability when organizations assign responsibilities, create plans and dedicate resources. SPI planning offers several advantages, including a common understanding of goals; a sequence of tasks with measurable objectives; coordination and monitoring of improvement and maintain the commitment of all participants in the SPI program.

The enterprises collaboration action plan includes three phases. The first phase is dedicated to an initial process specification. The second phase is basically an analysis and improvement of some projects based on the process initially defined and registers the new practices in order to create a new and improved version of the process. The final phase includes a feedback workshop. Each phase includes specific working methods and goals that we describe in this section.

Phase 1. Specification of the software development process: (1) analysis of existing documentation about the organization and eventually, if a process exists,

describes the existing process according to ProPAM; (2) kick-off meeting with all participants (project manager, process architect, projects teams, etc.) and (3) interviews to senior managers and project teams.

Phase 2. Analysis of process application in the organization projects: (1) tracking of some projects; (2) cooperative work in order to motivate team members for the best practices identified in phase 1 and towards this phase; (3) identify new entities (activities, products, roles, etc.) that had not been defined in the initial process and (4) compose a new process version with refinements and improvements detected.

Phase 3. Final workshops and final presentation of the new process or new process version with all participants.

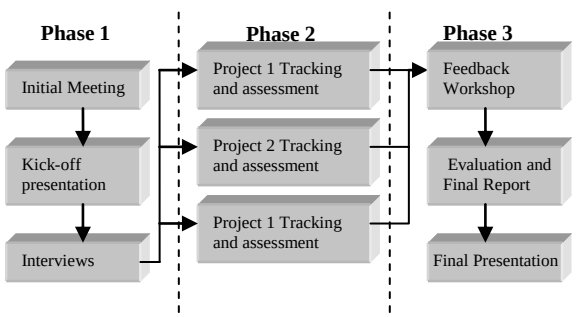

Figure 5: Enterprises Collaboration Action Plan

To support ProPAM, the methodology concepts are being integrated in a project management tool (ProjectIT-Enterprise) of a research project (ProjectIT) from INESC-ID Information Systems Group [26]. The main vision of the ProjectIT-Enterprise is to provide contributions mechanisms to development teams of average and great dimension, privileging on activities of process and project management and documents management, such as versions control, time management, quality or risk. The most recent version of ProjectIT-Enterprise already supports the fist and second components of the methodology – Process Definition Component and Project Definition Component. In the third component only some features are implemented like personal work management and a management issue for personal improvements proposals. These features allow team members to detect new improvements during projects iterations. Process versioning is not already implemented in the current application version.

4.1. PHASE 1

The interactions with organizations start with an initial meeting with senior managers which covers mainly the process documentation and the organization structure. Given that documentation and additional information, it became possible to start establishing an initial process according to the first component of ProPAM.

After this first approach, a kick-off presentation with all participants will cover mainly the description of the collaboration process and also a presentation of a draft of their process. This presentation is followed by senior managers and team members' interviews. The questionnaire includes questions according to the development process and projects considered.

4.1.1. THE BASE PROCESS

The initial specification of the development process is essential to start applying the proposed methodology. The team responsible by ProPAM, from now on called process group, creates the base process based on the available information. The Process Definition Component of ProPAM is used to describe the new process based on the PIT-ProcessM.

4.1.2. THE QUESTIONNAIRE

So far we have not identified any empirical study conducted with Portuguese practitioners to investigate what Critical Success Factors (CSF) play a positive role in the implementation of SPI programs. Although this work is not directly related with an empirical study about CSF, we believe that the CSF interview can also be useful in identifying important issues related to the implementation of SPI.

The questionnaire presents a list of possible CSFs which tends to limit them to those CSFs reported in existing international studies. However, those studies describe real life experiences on SPI implementation and we believe that results are also applicable in the Portuguese software development context. The interview often presents the initial occasion for the process group to interact with managers and others team members on the information that might be useful to them.

The questionnaire is structured in two sections. The questions regarding 'Human Resources and Competences' reveal team workers skills and future improvements in this area. The questions under "Process and Project Management" considered process focus and process improvement. The results from this questionnaire constitute the initial information to propose new improvements and practices to the development process.

4.2. PHASE 2

In the second phase, the ProPAM is applied in a project lifecycle. If managed effectively, the continuous changes and iterative cycles can motivate people to share ideas and experiences, try out new practices and

work together to reach challenging goals. SPI incremental natural requires long-term commitment for participants, but they must keep sight of the original goals.

The groups involved in SPI consist on the software development team, project manager and the process group. Here, we describe the competences of each one of the actors from these groups.

Project teams are in a central role in SPI, their collaboration within the project in the adequate definition, planning and improvement of the project activities are essential to detect changes that are candidates to SPI. The project team must be motivated to adapt the new practices in the next project iterations. The main goal must be optimize daily personal work and, consequently, to define a better process.

The main goal of a process group is to improve the process. The aim is to use knowledge from changes in the project iterations in order to adapt the base process. One of their tasks is to filter and analyze the process knowledge from the project. Observing just one iteration, the process group studies the candidates' practices to improve the process which feedback is eventually used to create a new version of the based process. This information will be applied in an evaluation period over the following project iterations.

The project manager is responsible for monitoring and detecting the project changes proposed by team members. The project manager has an important role in promoting the communication between all the groups involved in SPI. As we saw, the continuous collaboration between the project team and the process group implies a new set of cooperative activities that are necessary to the SPI effort. This additional work must also be considered as project activities and consequently as new practices to the improved process. For the project management functions, the improvement actions concentrate on common methods and practices across the complete development project, this comprises particularly risk management, project planning and tracking, quality assurance and configuration management.

The Project Definition Component is used by the project manager to define the project. This includes to create the project plan and to inform each project team member about their roles, activities and workproducts produced and consumed by their activities.

Iteration workshops are essential to incrementally detect and analyze the proposed improvements. These break points are important to discuss the current problems and obtain individual opinions from the project team. Although this is not a assessment period, the project manager must be sure that progress is made, needs to follow-up goals regularly to ensure that the project really strive to achieve the goals. This should be done regularly during the project and not just afterwards because then it is too late to adjust eventual issues.

4.3. PHASE 3

The process improvement assessments cover mainly the practice process. However, the proposed methodology is based on projects data, so project final results must be taken into account. The results of each project assessed are discussed with team members in a feedback workshop. This is followed by producing a final report with detailed results about the new process achieved and the final results of the assessment SPI program. In the end, these results are presented to all participants.

ProPAM is driven by small improvement iterations each with its data or outcomes. They need not all be good - but there always have some results. Those involved can learn from, and are motivated by the successes, and can learn from the mistakes. The feedback workshop is important because it is concentrated on the experiences from the entire project instead on data collected in each iteration. This kind of workshop is valuable to comment and improve the way this methodology is applied in organizations.

The final document presents a standardized result structure. For each discipline, several new practices are obtained which includes new activities, workproducts and roles. Eventually, this can comprises a maturity level for the overall process. Detailed verbally expressed findings as well as recommendations and proposals for improvements supplement the result structure. These data are essentially the result of the feedback workshop.

In the end of the project, a final presentation that includes also senior managers is dedicated to present the final results of the studies and also to turn participants more confident in their work.

5. CASE STUDY

The case study is applied in an organization in which CMM or other SPI model never had been applied. The organization is an international Software house located in Portugal and it has with two sectors: (a) Commercial Sector and (b) Software Sector. The software sector is a small team with 25 persons and the Commercial Sector has 10 persons. The major goal of the Software Sector is to develop and supply high quality business software products and services and also is subcontracted by other organizations. The organizations acts in three different areas but they don't know the characteristics and main differences in the development process of each area. The

Commercial Sector also had some problems. In particular, the communication between the two sectors presents some weakness. All these factors contributed to increase projects problems. In other words, no software development process was identified, no repeatability could exist.

The changes described here started when ProPAM was applied in three projects of different areas that were developed in parallel, but this paper evaluates only the impact that the methodology had in one project. The project concerns the development of an Information System with a small participation of the Commercial Sector. In practice, this means that the case study focuses in discovering new and existing practices that are important to have a description of the process independently of the project area. The main objective is to find if ProPAM is a good methodology to define and improve organizational development process, eventually according to standards like CMMI or SPICE. The particular objectives of the methodology components are: (1) create at least a repeatable process; (2) has project management aligned with process management; (3) prepare project teams to continuous process improvement.

The action plan described in the previous section is applied in order to test the methodology proposed in this paper. The case study took place in three phases according to the action plan. The first phase was dedicated to interviews and to an initial process specification. The second phase was concentrated in following team members' work. Data was collected considering the work planned for each team member, the work they really perform, new findings, recommendations and proposals for improvements. Each element filled out a daily time sheet, where the activities carried out were specified in agreement with the project work plan and the starting and finishing times were register. ProjectIT-Enterprise was the application used to collect these data. Monitoring and advisement became a systematic activity of the process group, who made sure that changes and improvements were register in the project. In the final phase all data was evaluated and final results were presented and discussed with team members and senior managers.

5.1. INTERVIEWS

Five interviews were conduced. The participants in these interviews fall into the following categories:

- The first group was composed by senior managers and project managers (referred to as managers);
- The second group was composed by analysts, designers, programmers and testers (referred to as team members).

The questions were both open and close-ended. The interviews took place in the interviewee's office. Additionally, the participants were informed about the nature of the research though a briefing kick-off meeting, already mentioned in section 5. The following steps were followed during interviews:

- First of all, the objective of the interview was described with some explanation of the research being undertaken.
- In case of a organization that already as performed a SPI program, some questions were made about the participant experience and knowledge in SPI implementation (not applied in this case).
- Questions were then asked about the CSF that they considered important for this SPI initiative.
- After having providing all the answers, the participants were then asked to provide their opinion about new practices or problems that they face in recent projects.

5.2. PROCESS SPECIFICATION

ProPAM proposes PIT-ProcessM, a meta-model to specify software development processes. The approach presents two views (dynamic and static) in order to show a temporal perspective and a discipline perspective. Here we just present the discipline perspective, since in the beginning of the projects we didn't know exactly how many phases and iterations characterize this process. By specifying the workflow in all process disciplines, the organization could take a step away from depending on individual, towards a predictable and schedulable working environment. Additionally, this first specification was taken as a baseline for further comparisons.

The static view of the process is organized in five disciplines: project management, analysis and design, development, tests and deployment. The activities, workproducts and roles used in these disciplines are described here.

The Project Management discipline (figure 6) includes activities involving the initial commercial proposal, project meetings, schedule activities, monitoring and controlling the project. The roles responsible to perform these activities are the service manager, the sponsor (client) and the project manager. The workproducts produced and consumed are the Commercial Proposal, the Project Draft and initial Plan and also meeting notes.

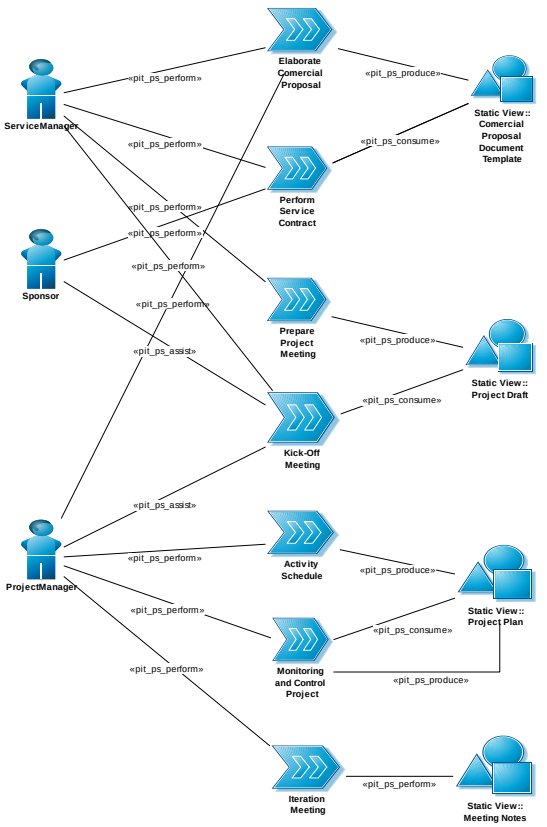

Figure 6: Project Management Discipline

Analysis and Design discipline (figure 7) includes the System Analyst responsible for collecting the system requirements and build the system model (Architecture Solution Design). The Client Project Manager as an important role in the presence of new requirements that consists in elaborating an issue or change request document. The System Analyst as to react and manage the changes sends by the Client Project Manager.

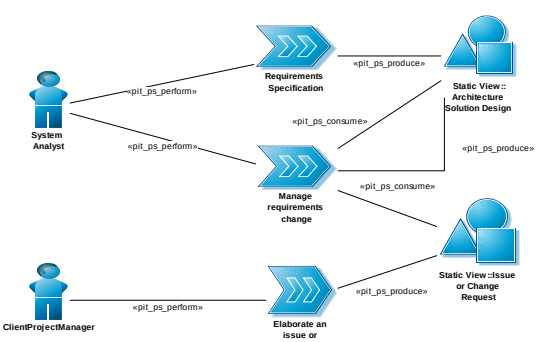

Figure 7: Analysis and Design Discipline

Considering Development discipline (figure 8), the Architecture Solution Design will be communicated to the Programmers who will turn this model into code and SQL documents. Additionally, Programmer will report their daily work in Timesheets.

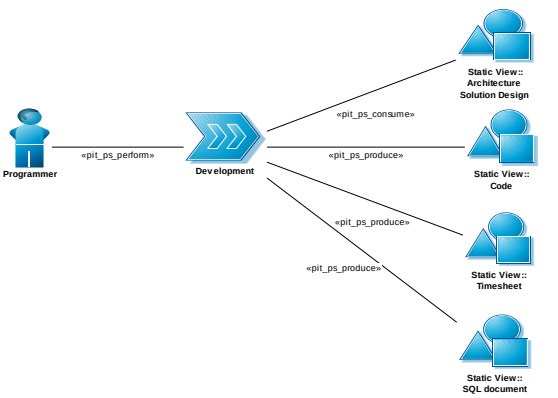

Figure 8: Development Discipline

The Test discipline (figure 9), once the application has been completed, it's time to test the application to determine its suitability against the original list of requirements. Testing is handled by testers and System Users throughout the course of the applications development. It can be a complicated process as Testers handle unitary testing of the code, as well as system tests. Once the application is sent to testers, they can identify exactly which files make up the application release, and therefore which versions of the files are likely to have bugs and faults that need to be addressed. Users also perform client test and need to record defects identified in the applications producing a final report called Bug Report.

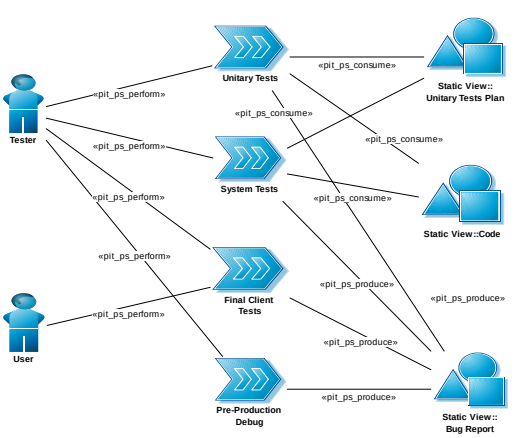

Figure 9: Tests Discipline

From an administration point of view, the

Deployment discipline (figure 10) includes the preparation of the project presentation and also the final exposition to the Sponsor (client) performed by the Project Manager. If the Sponsor accepts the presented system, a project acceptance document is assigned.

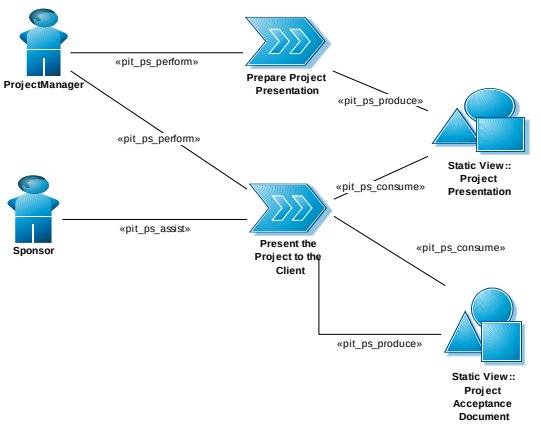

Figure 10: Deployment Discipline

Although some doubts and problems were identified in that moment, they were not immediately processed since they constitute improvement practices used to validate the proposed methodology.

5.3. Project Definition and Tracking

The project was divided in three phases in order to provide better management control and appropriate links to the ongoing operations of the organization. Each phase ends up with a project deliverable and a review of project performance in order to determine the main problems within the project. Phases are composed by several iterations and in the end of each one, an iteration workshop is realized to propose corrective actions and consequently process improvement.

The initial project plan is based on the process presented in the previous section, but as mentioned the project follows the dynamic view based on phases and iterations. Figures 11, 12 and 13 presents the diagrams with the workflows of the first iteration from each of the three phases identified in the beginning of the project. Figure 11 corresponds to the elaboration and presentation of the proposal to the Sponsor. If the Proposal is not accepted by the Sponsor, the Service Manager and the Project Manager will repeat these activities in a new iteration until the proposal is accepted or the Sponsor ends the project.

5.3.1. Contract Proposal Phase – Iteration 1

In the first iteration from the Contract Proposal Phase, the System Analyst and Project Manager reported additional activities, such as: an initial kick-off meeting with the Client Project Manager, interviews with Managers from different areas concerning the project, details about the commercial proposal document. The commercial proposal document is composed by three parts: system context (system architecture), project planning and software products.

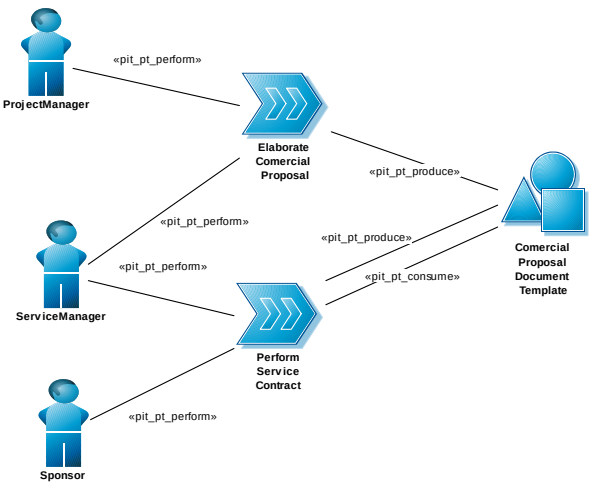

Figure 11: Iteration 1 from Contract Proposal Phase

Iteration 1 from Contract Proposal Phase based on the reports submitted by team members and in the information from the iteration workshop, the process group identified several problems: (1) Managers from client areas were not really motivated to participate (some of them change the meetings date several times); (2) the organization didn't has a knowledge repository about different architectures. He had to do some research in order identified the best architecture. Previous experience from others members from the organization could be considered if they had a common reposition. (3) The Project Manager spent a lot of time creating the project plan. He also spent additional time confirming project schedule with other team members.

In the end of the iteration workshop, the process group introduced some changes in the process. These improvements included: a knowledge reposition about information systems architectures, guidelines in software development, testing, etc; the project plan will be created based on the development process. In order to get more confidence in time schedule and costs, a historical repository about process entities must be created and updated based on projects data.

The process group also made a comparison between the project plan based on the process specification and

project plans created by the Project Manager. Table 1 presents the main features, advantages and disadvantages of the new approach.

Table 1: Project plan features

Features	Previous Project Plan	ProPAM Project Plan
Process lifecycle	No	Phases and Iterations
Discipline based	No	Yes
Process activities	No	Yes
Workproduct identification	No	No
Phase planning and control	No	Yes
Methodological approach	No	Yes

Now the main advantage is the effort saved in identifying the proper needs and structure for organizing and managing the project, once the project plan is now based on the process already defined. Another advantage is that the project plan reflects the process refinements over time in order to maintain the applicability, usability and acceptability of the process. The project plan is organized to allow problems identification structured in disciplines, phases or iterations. However, most project management tools does not support information flow, a concept essential in project execution support in which document routing and information-flow-based notifications are key. We therefore extended ProjectIT-Enterprise to support these features through definition of input and output workproducts in project activities. So, if some problem is associated to a specific workproduct, it is easier to identify the workflow of activities that probably must be repeated.

5.3.2. IMPLEMENTATION PHASE – ITERATION 1

Figure 12 presents the first iteration from the Implementation Phase. Once the proposal is accepted by the Sponsor, team members are associated to the roles identified in the process and corresponding activities are scheduled to them. This activity is performed by the Project Manager. This iteration includes an initial kick-off meeting and more detailed requirements specification. The first iteration included essentially analyses and design activities, but some implementations and tests were performed. In the end of the iteration, several situations were discussed in the workshop. Some problems concerning all disciplines involved in this phase were analyzed. As an example, we present a scenario that includes several actors and it's related with the quality of the information system developed.

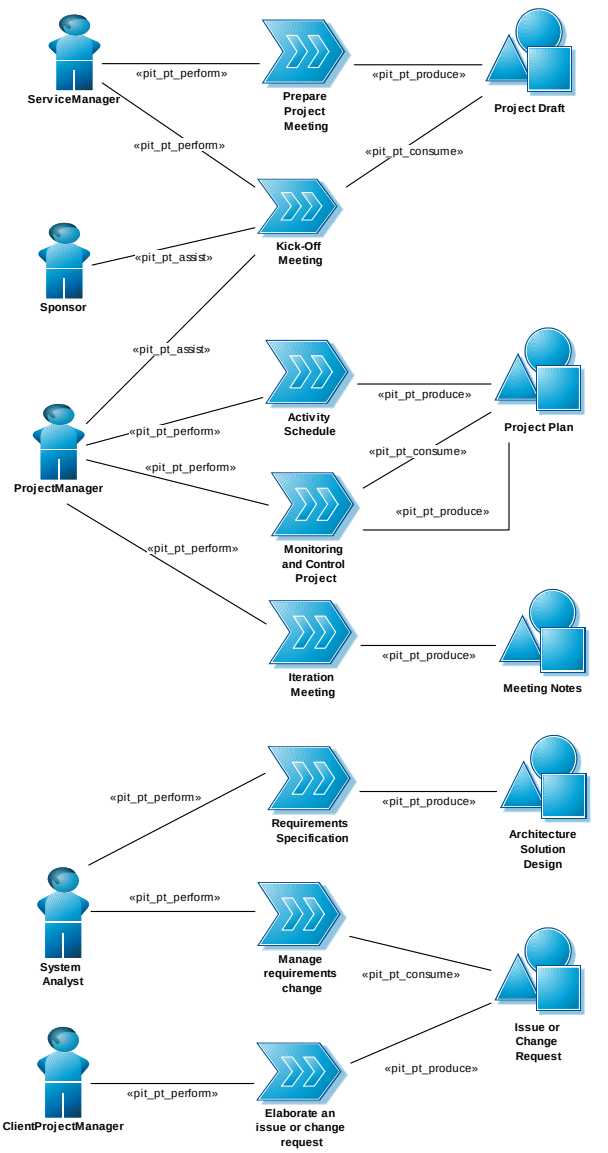

Figure 12: Iteration 1 from Implementation Phase

The problems associated with this scenario were:

Problem 1: The Programmer wrote they code but they didn't write the test first. Sometimes they also neglected tests.

Problem 2: The Programmer sometimes gets stuck or frustrated and needed help to found a solution. The Project Manager or a more experience programmer stopped their work and gave some guidelines about how to solve the problem.

Problem 3: Client users just performed their test

after the applications was delivered for tests.

Problem 4: System tests must be performed by other team member and not by the programmer that wrote the code.

The process group proposed improvements in the process in order to solve these problems.

Solution 1: If Programmers write the test first, the test drives the code. As a result, the code will look remarkably different and much simpler.

Solution 2: Pair programming is an alternative approach but each actor should switch roles frequently, changing from the driver (code writer) to the partner and so on. This approach also involves design decisions, less chance of both actors neglected test, spreads knowledge throughout the team and frequent code reviews.

Solution 3: Client Users must write their own tests. Client tests tell the team whether the system does what Client Users want it to do.

Solution 4: Improvements for quality assurance cover a common quality assurance plan for the system and reviews for all work products using harmonized review methods.

5.3.3. DEPLOYMENT PHASE – ITERATION 1

The number of iterations in the Deployment Phase (figure 13) depends on problems identified when the client organizations starts using the system. Until the Sponsor has formally accepted the project's deliverables and end products, the team continues performing their activities. The Project Manager controls and monitories the project, he also realizes some meetings in order to know about project features and restrictions to communicate to the Sponsor.

The case study aims to study how this methodology suits for self-adapting and improving the practices during a software project. Our approach to process improvement includes controlling and monitoring the project and iteration workshop to discuss and present the new practices to introduce and test in the future project iterations. The team members maintained daily records of their work and about positive and negative perceptions. The process group made several proposals of applicable practices based on team members' iteration reports. In the iteration workshop, they decide after discussion which practices are usable and how. During the project quantitative and qualitative data was collected considering: a) effort spent on workshops; b) quantity of new practices found; c) quantity of proposed and actual implemented process practices.

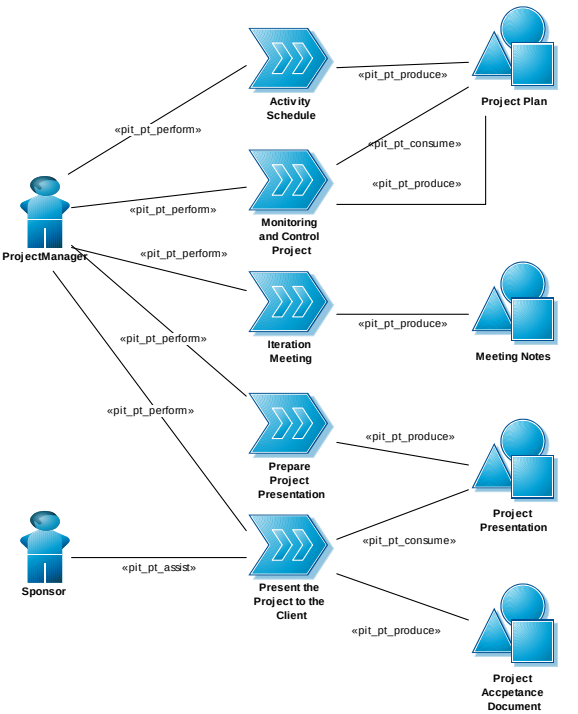

Figure 13: Iteration 1 from Deployment Phase

In the iteration workshop all team members' reports were considered as equal and all of them were discussed in the final iteration proposal presented by the process group. Each team member had to list his practices in order to verify his new schedule of activities. Finally, in the end of the workshop, the previous proposal was revised to find out what improvements had actually taken place and which ones were not implemented for one reason or another.

The length of iterations was different in the three phases. The project started with three two weeks iterations to the Contract Proposal phase and finished with two, two weeks iterations to the Deployment phase. The Implementation Phase had five four weeks iterations. This paper involves analysis on the first iteration of the three project phases.

5.4. SPI ASSESSMENT

The central goal established during the case study was to specify and improve the organizational software development process. The aim was to use the methods from the ProPAM in order to adapt the initial base process to better suit the software development areas of the organization and deal with any immediate weaknesses in the development practices. However, short time SPI is not always possible since changes do not occur immediately. It takes time for a new habit to be accepted and utilized [27]. As an example, it can take

years to move from a maturity level to the next [28]. Therefore, improvement can only be achieved in small steps, and all that already works well has to be maintained carefully. The idea was never to achieve a higher maturity level (level 2) in one attempt, but to take a step in achieving better quality and efficiency process.

Considering the lack of the process historical data, since the previous information from projects hadn't detailed data about process elements (phases, iterations, disciplines, activities, roles, etc), the current process was analyzed and taken as a baseline for future comparison. A quantitative evaluation in terms of the proposed metrics is not possible. Thus, we just performed a quality process assessment and generate some improvement advices.

The case study revealed the importance of continuous collaboration between the process group and team members. Different cooperative activities were identified as necessary to benefit the team as well as the process group in their SPI efforts. The amount of team members' proposal decreased towards the end of the project, as a consequent the time spent on iteration workshops to discuss SPI actions for the next iteration also decreased.

In the beginning of the project the team members productivity was at its lowest when the new methods and tools were still new to them and the initial activities were highly time consuming. The activities time were underestimated in the initial iteration but significant iterative improvement were observed influenced by factors, like learning through monitoring and interpreting the data of previous iterations. Along three iterations, by monitoring the actual productivity of each team member from previous iterations, the team was able to tune planned velocity for the scheduling of the next iteration. The ability to estimate productivity and activities time increased towards the end of the project.

The main goal of this paper was to analyze the effectiveness of the proposed methodology in SPI. The immediate goal of the process group was to analyze and transfer knowledge from project team members to the development process. The resulting improvements proposed could be separated in two groups: 1) practices immediately introduced in the process and 2) practices accepted at organizational level but not implemented in a short term.

On the average, only 45% of proposed improvements were implemented immediately and without organizational support. Thus, 55% of the new proposed practices were not implemented in the project lifecycle. These improvements were delayed since require additional efforts and senior managers decided not to give priority to these changes. Although proposed by the process group, the historical reposition with associated mechanism to analyze, transfer and disseminate project data to the process were not implemented due to organizational decisions. Moreover, 25% of the improvement planned and executed failed or were just postponed by any reason.

6. CONCLUSIONS

This paper proposes a new methodology for process improvement. The final outcome of the research is a methodology to assist in the definition and improvement of software development processes based on projects data. As shown in section 4, the methodology comprises a PIT-ProcessM meta-model, a PIT-ProjectM meta-model, a versioning meta-model and a Project Iterations approach. Currently, standard process meta-model, e.g. SPEM, is not suitable for SPI, since his main goal is on process specification without any consideration regarding project management or analysis. ProPAM concepts are being integrated in a project management tool (ProjectIT-Enterprise) of a research project (ProjectIT) from INESC-ID Information Systems Group [26].

ProPAM was applied in a case study to help the organization improve its software product development process. Results on using the improved process were gathered and analyzed. The case study reported here provides information about the utility of the methodology and also will contribute to some refinements on the basis of the experience and suggestions with industry practitioners. The results of the case study reveals that the proposed methodology – ProPAM- is a good approach to help project teams in improving the organizational software development process based on their daily work.

The contribution of the case study is its empirical findings because there isn't a lot of empirical data about the practical functionality of process models. The findings of the case study contribute to encouraging practitioners to start improving their process. The limitations of the case study were the absence of historical data to allow an assessment of the effectiveness of the improvements in the process. However, the case study does not offer extensive enough data to draw any generalizations, some conclusions can be brought forward for further evaluation.

Several conclusions could be made based on the data of the project: (1) the ability to estimate productivity and activities time increased towards the end of the project; (2) the higher productivity from team members in the end of the project somewhat indicates the learning of using the new methods and tools, but it also correlates to the increased satisfaction of the team members.

Traditional SPI takes too long to deliver benefits to software developers, managers or the organization. Traditional SPI planning often schedules improvements in terms of many months or years before useful results or objectives are achieved. Small or rapidly changing organizations will, quite reasonably, loose patience at these timescales, or may be reorganized or restructured before they are achieved. ProPAM is a methodology to process improvement based on projects iterative proposals. Many benefits arise from small and simple improvements, made quickly. The rapid visibility of the SPI actions and the concrete possibility to influence the working practices also increase the satisfaction of the project team.

The case study reveals that SPI isn't trivial, some improvements need organizations decisions, additional effort and time consuming activities. Some improvements had been implemented in the project as suggested, but some of them failed. SPI must follow an iterative approach but results aren't always immediately. 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. Also important is the need to repeat this study in other projects in order to draw any generalizations.

REFERENCES

[1] O. Salo. Improving Software Development Practices in an Agile Fashion. *Agile Newsletter 2/2005*, Agile-ITEA, pp. 8, 2005.

[2] B.C. Hardgrave, D.J. Armstrong. Software process improvement: it's a journey, not a destination. *Communications of the ACM* , 48(11), pp. 93-96, 2005

[3] Software Engineering Institute. Capability Maturity Model for Software (CMM), Version 1.1, Carnegie Mellon University, 1993.

[4] Software Engineering Institute. Capability Maturity Model Integration (CMMI), Version 1.1, Carnegie Mellon University, 2002.

[5] SPICE Project. Software Process Assessment Part 2: A model for process management, Version 1.0, 1998.

[6] P. Kuvaja, J. Simila, L. Krzanik, A. Bicego, G. Koch, S. Saukkonen. Software Process Assessment and Improvement: The BOOTSTRAP Approach. Blackwell Publishers, 1994.

[7] K. Beck et al. Manifesto for Agile Software Development. http://www.agilemanifesto.org/principles.html, 2006

[8] J. V. Vandeville. Organizational Learning through the Collection of Lessons Learned. *Informing Science*, vol. 3, pp. 127-133,2000.

[9] V. Gruhn, J. Urbainczyk. 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*, Kyoto, Japan, 1998, pp.13-21.

[10] B. Curtis, M. I. Kellner, and J. Over. Process Modeling. *Communications of the ACM*, vol. 35, pp. 75-90, 1992.

[11] H. Krasner, J. Tirrel, A. Linehan, P. Arnold, and W.H. Ett. Lessons learned from a software process modeling system. *Communications of ACM*, vol. 35, n.9, pp. 91-100, Sept. 1992.

[12] G. A. Bolcer, R. N. Taylor. Endeavors: A Process System Integration Infrastructure. *International Conference on Software Process* (ICSP4), Brighton, U.K., 1996.

[13] V. R. Basili, G. Caldiera, G. Cantone. A Reference Architecture for the Component Factory. *ACM Transactions on Software Engineering and Methodology*, 1 (1). pp. 53-80, 1992.

[14] S. Henninger, J. Schlabach. A Tool for Managing Software Development Knowledge. *Third International Conference on Product Focused Software Process Improvement*, Germany, September 2001, pp. 182-195.

[15] 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.

[16] H. Krasner, J. McInroy, D.B. Walz. Groupware research and technology issues with application to software process management. *IEEE Transactions on Systems, Man and Cybernetics*, 21(4), pp. 704--712, July/August 1991.

[17] N. L. Kerth. Project Retrospectives: A Handbook for Team Reviews, Dorset House Publishing, April 2001.

[18] A. Cockburn. Crystal Clear: a Human Powered Methodology for Small Teams, Addison Wesley, November 2004.

[19] 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, pp. 4-12, 2002.

[20] O. Salo. Improving Software Process in Agile Software Development Projects: Results from Two XP Case Studies. *30th EUROMICRO Conference* (EUROMICRO'04), Euromicro, 2004, pp. 310-317.

[21] Object Management Group. Software Process Engineering Meta-model Specification, Version 1.1, January 2005.

[22] 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.

[23] M. E. Loomis. Object Versioning. *Journal of Object-Oriented Programming*, January 1992.

[24] Y. Mashiko and V.R. Basili. Using the GQM Paradigm to Investigate Influential Factors for Software Process Improvement. *J. Systems and Software*, vol. 36, no. 1, pp. 17–31, 1997.

[25] J.H. Iversen, L. Mathiassen, and P.A. Nielsen,. Risk Management in Process Action Teams. *Improving Software Organizations: From Principles to Practice*, L. Mathiassen, J. Pries-Heje, and O. Ngwenyama, eds., Addison-Wesley, pp. 273–286, 2002.

[26] A. R. Silva. O Programa de Investigação Project-IT", version 1.0, October 2004.

[27] T. Packard. TQM and Organizational Change and Development. *In Total Quality Management in the Social Services: Theory and Practice*. B.Gummer and P. McCallion, editors, Albany, NY, Rockefeller College Press, 1995.

[28] W. Hayes, D. Zubrow. Moving On Up: Data and Experience Doing CMM-Based Process Improvement. *Technical Report* CMU/SI-95-TR-008, August 1995.