5.2 A Software Process Improvement Solution for Small and Medium-Size Enterprises

Authors

Jose A. Calvo-Manzano, Gonzalo Cuevas Agustín, Iván García Pacheco, Tomás San Feliu Gilabert, and Ariel Serrano

Abstract

The focus of this paper is to outline the main structure of an alternative software process improvement method for small- and medium-size enterprises. This method is based on the action package concept, which helps to institutionalize the effective practices with affordable implementation costs. This paper also presents the results and lessons learned when this method was applied to three enterprises in the requirements engineering domain.

Introduction

During the last few years, several software process improvement (SPI) models have been developed to increase the quality and productivity of software. Models, like IDEAL [McFeeley 96] or ISO/IEC 15504 [ISO/IEC 05], have been useful to initiate a software process improvement effort in many organizations. However, such models are oriented to large enterprises and their implementation frequently implies a high cost that is not affordable to small and medium-size enterprises [Cuevas 98].

This paper presents an alternative SPI solution for small and medium-size enterprises. This solution is called MESOPYME and is based on the Action Package concept [Fowler 97]. The experimentation of this package has been carried out on the Requirements Engineering domain and specifically on the process areas of Requirements Management and Requirements Development. These process areas have been selected because our investigations have found that most of the Spanish enterprises focus their improvement priorities on these areas.

The MESOPYME Method

MESOPYME has been developed as a SPI method that is focused on small and medium-size enterprises. Its structure has been divided in two parts. The first part is focused on process assessment and is based on a two-staged questionnaire that is a tool to determine the current state of the process [Cuevas 04]. The second part of the MESOPYME structure is focused on process improvement and uses the Action Package concept to establish and maintain a new process [Calvo-Manzano 00]. This paper only addresses the second part of the MESOPYME method and presents the Action Package main structure. Also, this paper presents the results obtained from method experimentation on the Requirements Engineering domain. This experimentation was carried out in three enterprises.

Action Package

The Action Package is structured as a set of components (organizational, management, and technical) that help get a solution for a specific software process domain. The Action Package is the mechanism that assists faster and affordable SPI program implementation for small and medium-size enterprises (SMEs). The main architecture of the MESOPYME Action Package is shown in Figure 31.

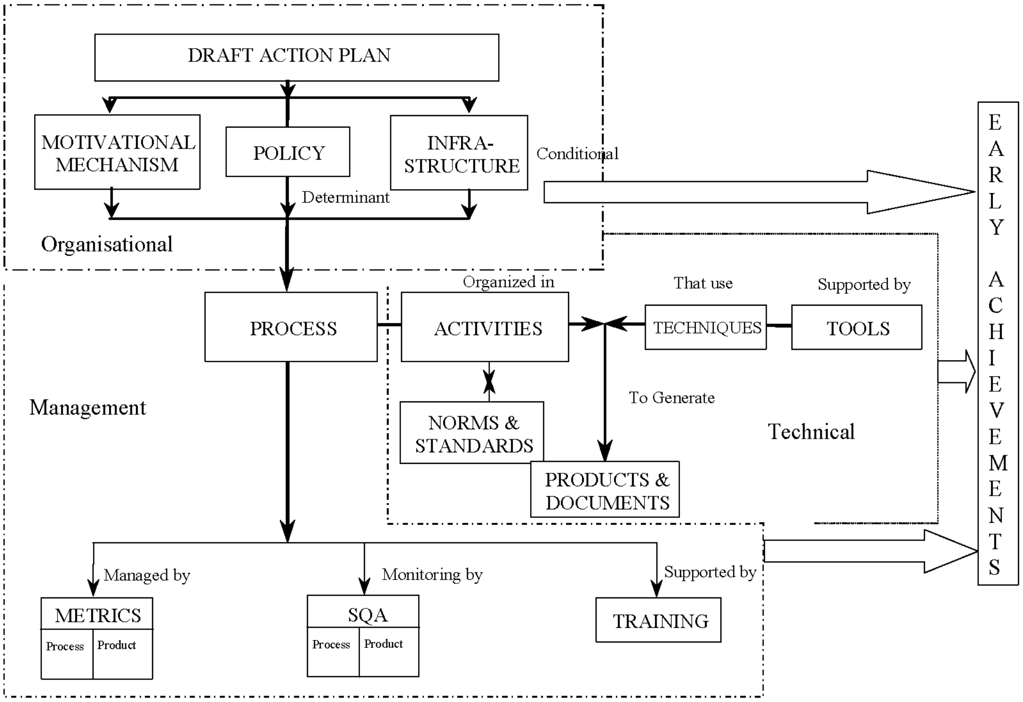

Figure 31: Action Package Architecture

- **Draft Action Plan**: This component contains a generic plan for the improvement project. The structure of the improvement plan helps to define with precision the context for the improvement action, its objectives, scope, and the specific tasks. The generic activities of the improvement plan include the following:

  1. define organizational structure
  2. establish the improvement context, with initial training in process, teamwork, and change management; conduct global and technical training in the action package
  3. define a short-term action plan
  4. review and adapt the action package

5. select the pilot project
6. conduct training to implement the techniques
7. run and collect measures
8. assess the pilot of the new processes
9. refine processes by having the pilot results in mind

- **Motivational Mechanism:** This component is used to get commitments from relevant stakeholders and to achieve process institutionalization. Human and organizational aspects are considered.
- **Policy:** Principles that drive the improvement strategy of the enterprise are stated explicitly. The package describes a guide of the policy for each process area. Also, some general rules, policy content, document format, practical cases, and measures to support the policy are provided.
- **Infrastructure:** For each process included at the package, an organizational infrastructure must be implemented. The roles and the responsibilities needed are established. A software process improvement structure based on the roles that people have to perform is proposed, independently of their position in the organizational structure.
- **Process:** For each process area included, the process activities are established. These activities are represented in a graphical way and are complemented by a detailed description. Any additional artifact (documents, templates, techniques and tools) are described in a matrix relationship with, if needed, the tasks involved in their execution, documents and/or the inputs used and the outputs generated, participant roles for each activity, and techniques used to carry out each stage of the process.

Examples of graphical representation corresponding to Requirements Engineering domain are shown in Figure 32 and Figure 33.

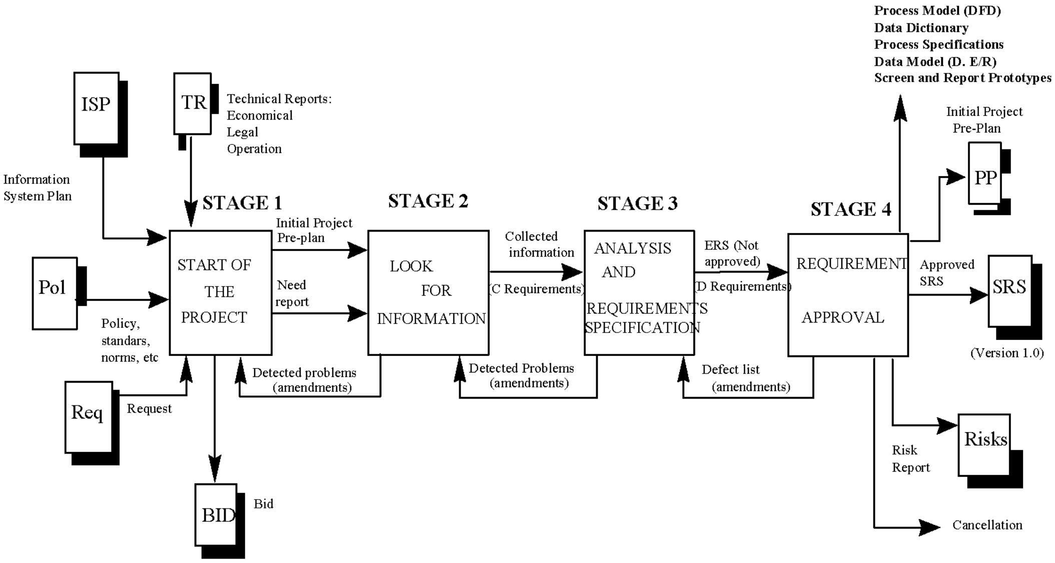

Figure 32: Requirements Development Process

In the case of the Requirements Development process area four stages have been defined. In each stage, activities involved, input documents and products, roles involved, organizational structures, output documents, and products and techniques to be used are specified.

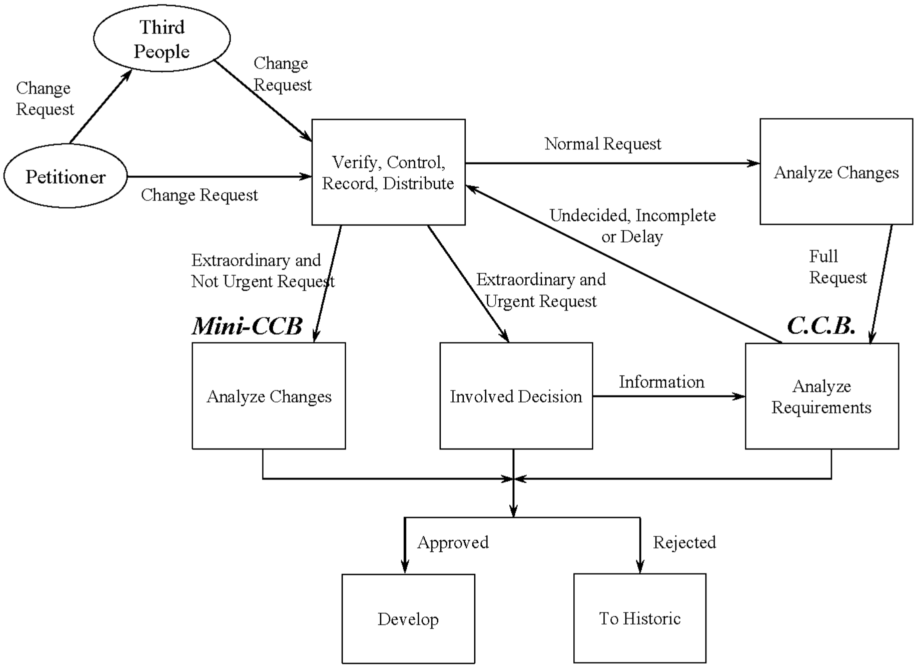

Figure 33: Requirements Management Process

In the case of the Requirements Engineering domain, roles defined are functional analyst, technical analyst, customer, change control board (CCB), project leader, quality expert, operation expert, methods and processes expert, system expert, and user.

- **Products and documents**: Global document architecture is defined to describe the detailed products formats, which will be obtained when applying the above processes. It is important to highlight that this architecture has common parts, such as general management documents for all action packages. The detailed designed formats include an agenda model that will be used in the meetings call and a change requests template that will be used to register and treat the change requests. An example of the Requirements Engineering document structure is shown in Figure 34.

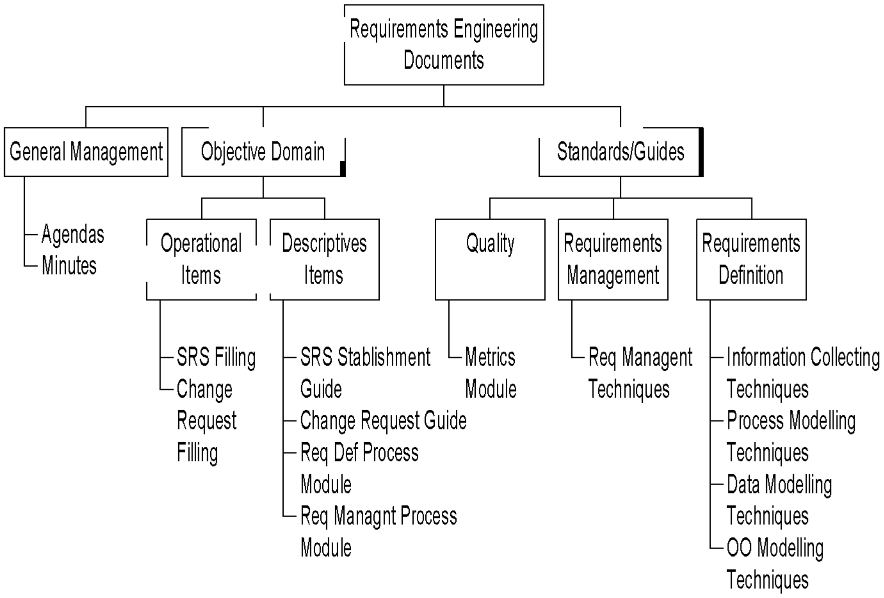

Figure 34: Structure of the Requirements Engineering Action Package "Products and Documents"

- **Techniques:** The techniques to be applied in the activities described in the process are defined. For example, in the Requirements Development process area, techniques to extract information such as interviews and JAD (joint application development) are established. Moreover, techniques to model functional behaviors following a structured or object-oriented approach are presented. Finally, inspections techniques are also defined.

- **Tools:** Due to continuous evolution and changes of the tools and the wide variety of the platforms that can be found in enterprises, the Action Package does not provide a tool list. However, the Action Package provides tool taxonomy to guide the selection of an adequate tool. In this way, when the process corresponding to an action package has been established, enterprises can select quickly and easily to get the more convenient tool for them.

- **Metrics:** To begin a successful software process improvement program, it is important to quantify it, based on the attributes which help us to understand issues that affect quality, opportunity, cost, process implementation rate. All of these issues are key in the decision making process. The action package contains a set of basic and calculated metrics. The metrics are selected according the business goals. There are tables that relate business goals with specific metrics. There are defined procedures to register, using data collect templates, and to analyze data of the measures.

- **Software Quality Assurance:** Reduced software quality assurance requirements to be used for processes and products are described. Guides in which objectives, organizational structure, and functions of the quality assurance structure applied to software development are described. Also, in order to assess objectively the process and standards, a set of metrics is defined.
- **Training:** Training consists of courses on SPI, team building, CMMI, and the specific software process area to be improved. Usually a general course on the process area will be provided, and the course will present an overview of the current technology and standards that are applicable to the process area. Also, specific courses to introduce the action package and courses related to the implementation of the techniques and specific tools selected will be defined.
- **Early Achievements:** There are improvement goals to be achieved in a short period of time from the beginning of the improvement project. Early achievements aim to obtain visible results for the whole organization, so they maintain the commitment of the organization to the improvement project. For example the early achievements defined in the enterprises where the requirements engineering action package have been tested include the following:
  - establish a requirement engineering policy
  - define roles and responsibilities
  - define roles and responsibilities of the Change Control Board
  - define a change request template
  - start the revision of the current requirement change process with users
  - define a checklist of potential impact areas, which can be analyzed with requirements
  - define a template to define requirement in new developments
- **Glossary**: A glossary is included to provide a common vocabulary to assure intelligibility and utility. The glossary contains the more common terms of the process improvement technology, as well as specific terms used.

The action package architecture helps to do the following:

- separate information into useful components for different purposes
- select the parts of the component that have interest
- carry out change management and improvements easily due to the established architecture in components
- provide access to required information in an easy way

Experimentation Results

The MESOPYME method has been applied on three SMEs on the Requirements Engineering domain [ESSI 97a, ESSI 97b, ESSI 97c]. Also, two external consulting companies carried out the implementation of the method. Table 21 shows the industrial domains in which MESOPYME has been applied.

Table 21: Industrial Domains of the Enterprises Involved in MESOPYME Experimentation

Enterprise Sector → Enterprise↓	IT	Production	Service	Others
E1				X
E2	X			
E3			X	

Table 22 shows the sizes of software development units and enterprises in which MESOPYME has been applied.

Table 22: Employees Distribution — Totals and Those Involved in Software Development Unit

Employees→	Enterprise				Software Development Unit			
Enterprises↓	1-10	11-50	51-500	+500	1-5	6-20	21-50	+50
E1				10.000			45	
E2			257					83
E3		25				12		

The results obtained in these experiences have been promising and provide evidence that MESOPYME method could be adequate in small and medium enterprises. Also these results have demonstrated that MESOPYME method could accelerate the implementation of software processes improvement. The results show that the smaller the enterprise, the less effort to adapt the package, and the more profit made.

With the MESOPYME action package, the external help could be reduced to a half or third, according to previous experiences obtained without the use of MESOPYME. The experience confirms that a typical improvement life cycle is implemented around six months, with an average external effort of 2.5 month-people. The obtained results are shown in Table 23.

Table 23: MESOPYME’s Obtained Results

Phase→	Total duration per phase (months)				Total Time (months)	Effort and cost (month-people y K$ USD)			
Method↓						Internal		External	
	Initial	Diagnostic	Establishing	Acting	Total months	M-P	K$USD	M-P	K$USD
MESOPYME	0.5	0.5	0.5	4.5	6	8.5	28.3	2.5	32.5

Conclusions and Lessons Learned

The lessons learned from the MESOPYME method experimentation on three enterprises include the following:

- First, the action plan that must be done by the working team. The final action plan is obtained with the customization of MESOPYME action plan.
- Second, the process definition procedures have examples and prototypes of process areas. The procedures helped reduce the time spent in defining the processes. The experiences in these enterprises have demonstrated that the procedures were useful and easy to adapt to the enterprise needs. Having the components in advance helped the enterprises focus their work and avoid spending time deciding to develop or not develop a procedure. They also helped the enterprises avoid spending a great deal of time that is produced in the initial phases of an improvement project due to the uncertainties associated with the new way of working.
- Third, the effort to understand the possible solutions proposed and its adequacy to the enterprise has been reduced. Managers, project leaders, and technicians have understood their activities and responsibilities easily by the use of action packages. With other methods, this activity could be done spending more time because the working team has to design a whole solution.
- And finally, MESOPYME quality assurance activities have helped to create a SQA group that was identified as a gap in these enterprises.

The principal contribution of MESOPYME method is that the modular architecture is a good solution to software process improvement in small enterprises.

The action package has a modular structure that defines different elements to be considered in the implementation. This structure is common to all process areas and its implementation, is accessible to SMEs, allowing a fast technology transfer. The results show that it is possible to implement a few process areas in five to seven months at an affordable cost.

The implementation of MESOPYME has achieved good results in the enterprises where was applied it and minimized their change resistance. It is important to highlight that

MESOPYME includes the coordination of three SPI relevant aspects: human, process, and technology. Finally, with the obtained results, it has been found that the Action Package might be a satisfactory software process improvement solution for SMEs at an affordable cost.

References

[Calvo-Manzano 00] Calvo-Manzano, Jose A. “Método de Mejora del Proceso de Desarrollo de Sistemas de Información en la Pequeña y Mediana Empresa,” Doctoral Thesis. Universidad de Vigo, España, July 2000.

[Cuevas 98] Cuevas, Gonzalo & Gil, Miguel Ángel. “Mejora rápida en los procesos de desarrollo de software,” *II Congreso Nacional de Ingeniería de Telecomunicación.* Madrid: June 1998.

[Cuevas 04] Cuevas, Gonzalo; Serrano, Alan; & Serrano, Ariel. “Assessment of the Requirement Management Process Using a Two-Stage Questionnaire,” 110-116. *Proceedings of the Fourth International Conference on Quality Software (QSIC2004).* Braunschweig, Germany, September 8-9, 2004.

[ESSI 97a] ESSI, Erosky. “SCOOP. SYMQUAD: System Requirements and Quality Assurance in a Distribution Company,” ESSI Project: No. 10344, March 1997.

[ESSI 97b] ESSI, Lantik S.A. “LANHOBEK: Client Relationship and Requirement Management Improvement,” ESSI Project No. 24155, March 1997.

[ESSI 97c] ESSI, G.O.C. “HOSPUR: User Requirements and Reuse in Health Management Systems,” ESSI Project: No. 24181, March 1997.

[Fowler 97] Fowler, Priscilla & Patrick, Mac. “Requirements Management Transition Packages.” P*roceedings of the Introducing Requirements Management Into Organizations Workshop.* CMU/SEI Transition Enabling Program, May 1997.

[ISO/IEC 05] ISO/IEC 15504. “Information Technology — Process Assessment,” International Organization for Standardization, 2005.

[McFeeley 96] McFeeley, Robert. IDEAL: A User’s Guide for Software Process Improvement (CMU/SEI-96-HB-001, ADA305472). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996.
http://www.sei.cmu.edu/publications/documents
/96.reports/96.hb.001.html