Software Process Improvement: MESOPYME Model and Method

Calvo–Manzano Villalón, Jose A.[1], Cervera Bravo, Joaquín[2] and San Feliu Gilabert, Tomás[3]

[1] Facultad de Informática, Universidad Politécnica de Madrid, Madrid, España
[2] Universidad Carlos III de Madrid, Escuela Politécnica Superior, Madrid, España
[3] Facultad de Informática, Universidad Politécnica de Madrid, Boadilla del Monte, España

Product quality stems from the quality of the process used to develop and maintain it. For this reason, organizations have developed models and methods to continuously improve their software process. At this moment, these methods are only applicable to large organizations due to the costs required to begin an improvement program.

In this paper, we present a new model with its associated method, both called MESOPYME. With this method, small and medium-sized enterprises (SMEs) can support software process improvement reducing the costs and the time of return on investment.

Keywords: software process improvement, software process assessment, improvement program

1. Introduction

Ever since the "software crisis" was "discovered" in the 1960s, customers of software systems have been concerned about the quality of software products. Many solutions have been proposed: structured methods, prototyping, computer aided software engineering (CASE), object-oriented methods, etc.. Currently, the adoption of software improvement methods is offered as the latest solution to the problem, L. Fernandez, J. A. Calvo–Manzano (1996).

Large organizations are improving their software process applying methods of continuous improvement (Bootstrap, SPICE, those based on the CMM developed by Software Engineering Institute (SEI) of the Carnegie Mellon University, etc.) which, actually, are not applicable to small and medium–sized enterprises (SMEs) due to the costs (financial, time and resource) associated with their application. However, SMEs cover more than 99 per cent of all enterprises in Europe, ESI (1996).

In this article, a new model with its associated method is presented. With this method, SMEs can support software process improvement, reducing the cost and the time of return on investment.

2. Models and Methods to Improve Software Process

At the moment, the main methods of software process improvement are:

- CBA IPI: CMM-Based Appraisal for Internal Process Improvement, D. K. Dunaway, S. Masters (1996).
- SCE: Software Capability Evaluation, P. Bymes, M. Philips (1996).
- Bootstrap, P. Kuvaja (1995).
- AFA: Action Focus Assessment, ISPI (1994).
- SPICE: Software Process Improvement and Capability Determination, A. W. Graydon et al. (1995); C. Mackie, T. P. Rout (1995).

Some general characteristics of these methods are:

- Every method uses a similar reference model to improve software process. This model is composed of the following stages (see Figure 1, which corresponds to the IDEAL model developed by SEI of CMU, used with the

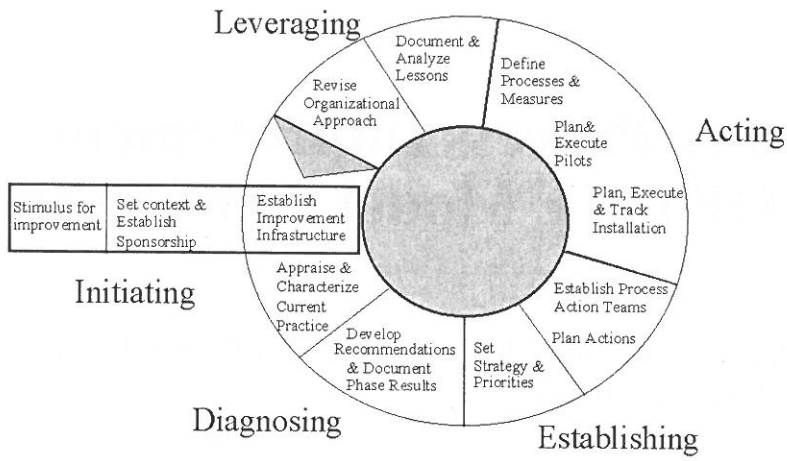

Fig. 1. The IDEAL Model for continuous improvement of software process

CBA IPI and SCE methods): initiate for the improvement, diagnose or assess current software process, establish or develop a plan of action to improve, act or implement the plan of action and, finally, leverage or institutionalize improvements.

- Every method, except AFA and SPICE, is focused on the initiating and assessment stages of their respective improvement models and, moreover, they are only applicable to large organizations. For other stages, the methods only give vague guidelines and consequently, an organization at Level 1 of maturity cannot establish and develop, on its own, a way to improve its software process, since in most cases it does not have enough capability to carry it out.

Other assessment methods of software process, such as Software Technology Diagnostic (STD) and TRILLIUM, are a derivation or adaptation of the previous methods.

- In the assessment stage, the methods use a similar software process model as a reference in order to obtain the strengths and weaknesses of the current software process. This reference model is the Capability Maturity Model (CMM), M. C. Paulk et al. (1993a), M. C. Paulk et al. (1993b), or some models derived from it.
- In aforesaid methods, it is necessary to consume a great deal of resources and time during the assessment and the first improvement cycle, and the return on investment is perceived between 18–24 months later. Specifically, in the AFA and SPICE methods (which are the only ones that support implementation of the plan of action), personal implication is greater than others because it is necessary to create work groups that design and implement each improvement to be introduced in the organization. These work groups imply a very high cost that SMEs cannot afford, because they have a very narrow margin between incomes and expenses. Therefore, they cannot sustain the investment and, moreover, the return on investment takes a long time. Table 1 shows costs in months, effort and money required by the previously applied methods.
- Existing methods have not incorporated graphical techniques to show results. Graphical techniques are very common in software engineering due to their high effectiveness in communicating information. In fact, existing methods show only textual information or global charts which do not provide enough visibility of the current situation of the enterprise.
- Due to the fact that the methods only carry out the assessment stage, metrics are not established to measure and track progress in the process improvement (nor in the AFA method), which is essential in order to know the implementation level that corresponds to the improvement.
- Some potential problems can appear with the improvement methods, such as:

Stages→ Methods↓	Initiate	Assess-ment	Plan of Action	Implemen-tation	Total Schedule (months)	Total Effort (month/person) External	Internal	Total Cost (ECUs)
CBA IPI	2	2	1	18	23	46	10	117000
SCE	6		–	–	6	2	6	17500
Bootstrap	1–2		–	–	1–2	1	1	11500
AFA	4	4	1	18	27	48	10	129000
SPICE	Trials				23–27			*

* Total Cost (ECUs) for SPICE method will be similar to AFA and CBA IPI methods

Table 1. Typical costs of the improvement/assessment methods

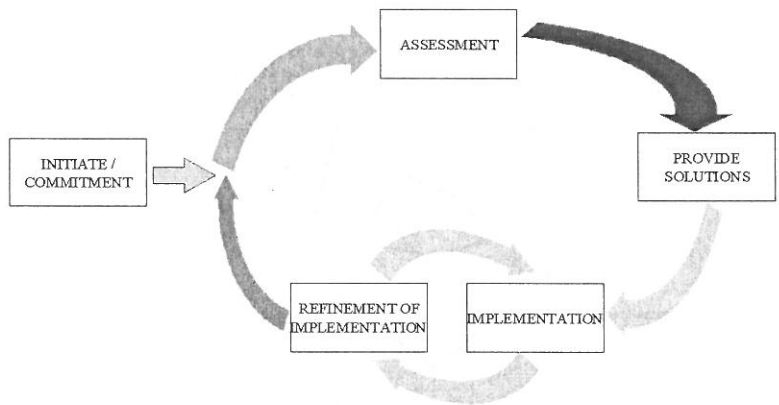

Fig. 2. MESOPYME model

- Staff demoralization if the assessment does not present concrete actions and results in a short time.
- Inadequate results if the assessment is conducted as an audit.
- Failure in the implementation of the plan of action if organization staff is not involved in the process of change.

3. MESOPYME's Model and Method

Our research has been focused on the development of a new process improvement model in order to create a method to assess and improve software process by reducing costs, time and resources in comparison with the existing methods. In this way, SMEs can begin an improvement program to consolidate their competitivity in their business area. The new method and the improvement model are called MESOPYME (Software Process Improvement in the SMEs; in Spanish: MESOPYME, *Método/Modelo de mEjora del proceso SOftware en la Pequeña Y Mediana Empresa*).

MESOPYME provides a new improvement model. The initial stage is similar to the above models, but other stages are completely different, so that SMEs have to make little investment and obtain a return on investment in a short time.

The MESOPYME model consists of a stage of commitment that is similar to that of existing methods, whereas the new stages are assessment (although there is a stage of the same name in the previous models, we consider this stage new because we use a quite different approach and development), providing solutions, implementation and refinement (see Figure 2). In this way, MESOPYME provides a different vision with regard to the current models and methods, and SMEs can sustain software process improvement with successful results. The following is a brief description of each of these stages:

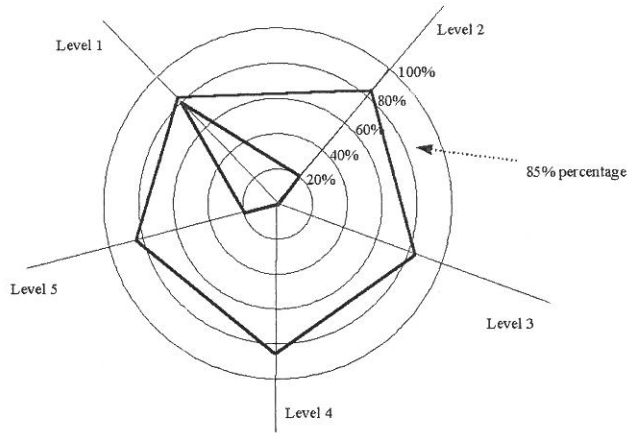

Fig. 3. Global maturity level

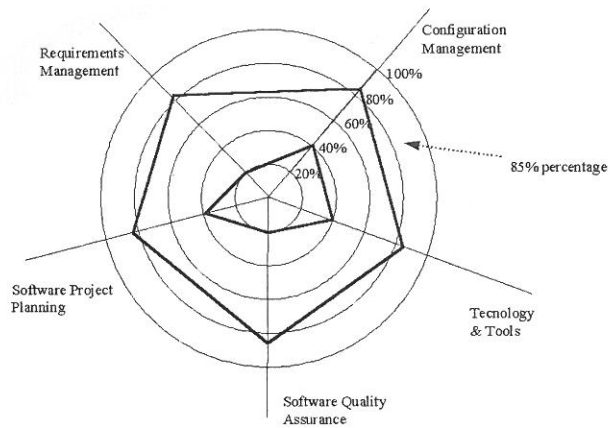

Fig. 4. Percentage weighted for Key Process Area of Level 2

- Stage 1: Commitment on the part of senior management. This stage has not been developed in the research because it is common to all existing models and is carried out by them in a similar way. Clearly, this is the most important stage because if we do not have the explicit commitment of senior management, it makes no sense to begin a continuous software process improvement.
- Stage 2: Assessment. The objective of this stage is to obtain the strengths and weaknesses of the organization using the CMM as a process reference model.

We use a maturity questionnaire based on those of the SEI and ISPI to obtain more visibility through the software process. The questionnaire has been adapted to European culture and questions related to technology and tools have been added.

Also, we have designed a communication model to show the results to those who are in charge of making decisions. This model enables an easy understanding of the problems and weaknesses to be improved. The communication model is composed of a graphical part based on Kiviat diagrams and column charts and a textual part which explains the graphical one. Kiviat diagrams correspond to the global maturity level of the organization (see Figure 3) and to the weighted coverage level for the key process areas of the maturity level (see Figure 4).

In those key process areas whose percentage weighted is lower than 85%, it is necessary to determine which activities have to be improved. For this reason, we have produced a column chart (see Figure 5) with all the activities in the key process area. Those activities whose percentage is lower than 85% are candidates for improvement because they are weak activities. Also, we calculate the media and typical deviation for the activities whose coverage is greater than 85%, in order to study possible discrepan-

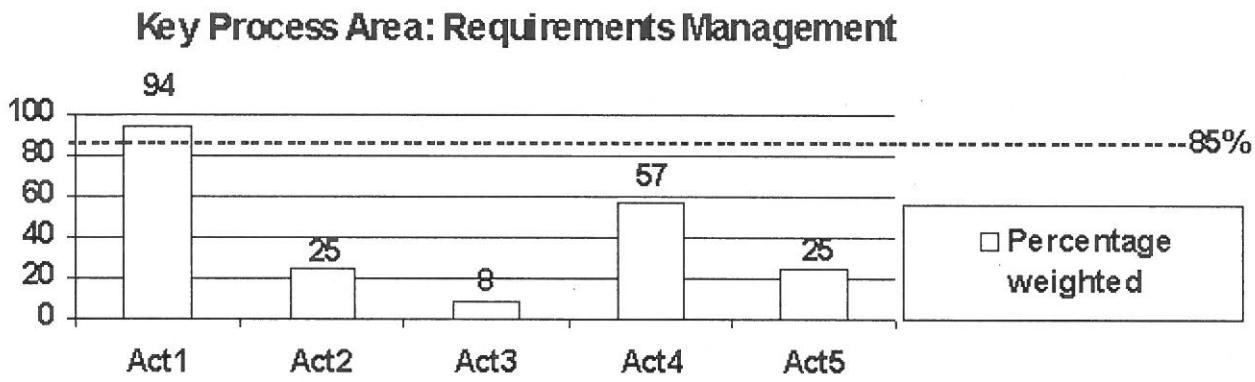

Fig. 5. Column Chart corresponding to the requirements management activities

cies in the questionnaire answers.

The next activity is to interview the persons who have been involved in the assessment in order to verify the answers given in the questionnaire and the results obtained by the assessment team.

Finally, by this procedure, the assessment team and senior management choose the key process areas to be improved.

- Stage 3: Providing solutions. With the selected improvements, we provide solutions that develop earlier (called "acting packages"). Sometimes, we will have to adapt these packages to particular business needs. An acting package is composed of the following modules:

  1. Introduction: scope and objectives of the package are developed along with the dependences on other packages. The package structure is also described.

  2. Policies: laws to govern operations of the organization are established.

  3. Infrastructure: structure needed to implement the acting package including roles and responsibilities is detailed in this module.

  4. Process: stages, activities and tasks needed to implement the package are described.

  5. Products and Documents: associated products and documents resulting from the activities above are described. A global structure is defined and composed of products and documents of: general management, domain of the objective and norms and guides.

  6. Techniques: techniques used in the related activities in the process are defined.

  7. Tools: a general taxonomy is presented in order to choose the most suitable tool for the organization.

  8. Metrics: a metrics model is described to satisfy goals of the organization and to help us to measure the progress of the package implementation and also the MESOPYME method.

  9. Training: the guide to implement the package is included.

- Stage 4: Implementation. In this stage, institutionalization of the package will take place in the organization by the SEPG.

- Stage 5: Refinement of the implementation. Once the package has been implemented, the organization has to take into account the learned lessons in order to refine the package. At this stage we use metrics values to refine the package if necessary.

In the MESOPYME method it is important to highlight:

- **Communication model**. People involved in the assessment stage have a better insight of their current software process, and they can notice more clearly what they want to achieve, taking into account the organization's business objectives. In short, they use this model as a help in making decisions.

- **Metrics model**. We have defined a generic metrics model to track the improvement evolution. This model can be applied to all issues that must be improved and provide us

MESOPYME	Total Schedule (months)	Total Effort (month/person) External	Internal	Additional Effort (month/person) External	Internal	Total Cost (ECUs)
Commitment & Assessment	1	2	1.1			25.000
Provide Solutions	5–6	0.5	4	0.25	2	

Table 2. Typical costs of MESOPYME

with some information about the process itself and about the metrics that we have applied.

- **Change management**. This aspect includes the identification, adaptation and institutionalization (technology transfer) of the standard improvement components that business requires, involving its own people and achieving results in a short time.

4. Conclusions

Nowadays, the acting package of requirements engineering (constituted by the requirements definition and management) is being implemented in some companies, one of which is sponsored by the CEC through a European ESSI project: SYRQAD, System Requirement and Quality Assurance in a Distribution company. Currently, techniques and process modules from the requirements engineering package are beginning to be implemented.

So far, we have verified the following reduction of costs (see Table 2): total schedule in the commitment and assessment stages has been reduced to 1 month; external total effort is 2.75, while internal is 7.1. At the stage of providing solutions, we do not take into account the effort of implementing the project itself, however we have added a column called *additional effort*. In this column we represent the effort due to the new improvement activities introduced that were not performed in the past by the organization. Total cost of MESOPYME is greatly reduced in comparison to CBA IPI, AFA and SPICE.

With this reduction in time, effort and money, software process improvement can be sustained by SMEs. We estimate that the return on investment will be perceived within a period of from 6 to 8 months as opposed to the current 18 months schedule of most of the methods currently used, not only because creation of work groups is made unnecessary, but also because we supply the improvement design.

New action packages related to key process areas of Level 2 and 3 of the CMM will be developed.

References

P. BYMES, M. PHILIPS, (1996) *Software Capability Evaluation Version 3: Method Description*, Technical Report CMU/SEI–96–TR–002, Carnegie Mellon University, Software Engineering Institute, Pittsburgh.

L. FERNANDEZ, J. A. CALVO–MANZANO, (1996), "Hacia la calidad del software a través de la mejora de procesos" , *Novática*, Septiembre-Octubre no. 123, pp. 35–42.

D. K. DUNAWAY, S. MASTERS, (1996), *CMM-Based Appraisal for Internal Process Improvement (CBA IPI): Method Description*, Technical Report CMU/SEI–96–TR–007, Carnegie Mellon University, Software Engineering Institute, Pittsburgh.

ESI (1996), "Software process: the new way to compete", Improve, The Newsletter of the European Software Institute, Issue 4, pp. 1–2.

A. W. GRAYDON, M. D. KONRAD, M. C. PAULK, (1995), "An Overview of SPICE's Model for Process Management" , *Proceedings of the 2nd International SPICE Symposium*, Brisbane, Australia, 1–2 Junio, ed. by T. P. Rout, pp. 27–34.

ISPI (Institute for Software Process Improvement), (1994), *Workshop on Software Process Assessment Training*, Iberia, Madrid, España, 14–18 Marzo.

P. KUVAJA, (1995); "Bootstrap: A Software Process Assessment and Improvement Methodology" , *Proceedings of the 2nd International SPICE Symposium*, Brisbane, Australia, 1–2 Junio, ed. by T. P. Rout, pp. 93–106.

C. MACKIE, T. P. ROUT, (1995), "Architecture of the SPICE Standards Suite" , *Proceedings of the 2nd International SPICE Symposium*, Brisbane, Australia, 1–2 Junio, ed. by T. P. Rout, pp. 15–25.

M. C. PAULK, B. CURTIS, M. B. CHRISSIS, C. V. WEBER, (1993a), *Capability Maturity Model for Software, Version 1.1*, CMU/SEI–93–TR–24, Carnegie Mellon University, Software Engineering Institute, Pittsburgh.

M. C. PAULK, C. V. WEBER, S. M. GARCIA, M. B. CHRISSIS, M. BUSH, (1993b); *Key Practices of the Capability Maturity Model, Version 1.1*, CMU/SEI–93–TR–25, Carnegie Mellon University, Software Engineering Institute, Pittsburgh.

Received: October, 1997
Accepted: December, 1997

Contact address:

Calvo–Manzano Villalón, Jose A.
Facultad de Informática
Universidad Politécnica de Madrid
Boadilla del Monte
28660 Madrid
España
Telephone: +34 1 3366924
Fax: +34 1 3366917
E-Mail: jacalvo@fi.upm.es

Cervera Bravo, Joaquín
Universidad Carlos III de Madrid
Escuela Politécnica Superior
Leganés
28911 Madrid
España

San Feliu Gilabert, Tomás
Facultad de Informática
Universidad Politécnica de Madrid
Boadilla del Monte
28660 Madrid
España

JOSE A. CALVO–MANZANO VILLALON He is assistant professor in the Computer Science Faculty of the Polytechnical University of Madrid. He is in train of obtain his PhD. in the field of software process improvement. He is author of several papers and books in the software engineering area and he has collaborated in several European projects.

JOAQUIN CERVERA BRAVO He has got a degree in Computer Science from the Polytechnical University of Madrid. He was working in the Facultad de Informatica of Madrid and actually, he is assistant lecturer at the Universidad Carlos III of Madrid, developing his PhD. He is author of several papers in the software engineering area.

TOMAS SAN FELIU GILABERT He has got a degree in Computer Science from the Polytechnical University of Madrid (UPM) and he is in train of obtain his PhD. in the field of software risk management. Currently, he is assistant professor in the Department of Software Engineering of UPM. He is author of several papers in the software engineering area.