Proceedings of the 5th WSEAS Int. Conf. on Software Engineering, Parallel and Distributed Systems, Madrid, Spain, February 15-17, 2006 (pp24-29)

Software process improvement for practitioners based on knowledge management tools

ANTONIO AMESCUA, JAVIER GARCÍA, MARIA-ISABEL SÁNCHEZ-SEGURA, FUENSANTA MEDINA-DOMÍNGUEZ
Computer Science Department
Carlos III Technical University of Madrid
Avda. Universidad, 30, Leganés 28911, Madrid
SPAIN

Abstract In order to spread the use of software process improvement programs and to make their use independent of organizations features, this work describes the results obtained using a knowledge based model and tool, as well as the proposal of using a patterns based solution implemented using a SPEM extension, in order to improve the efficiency of use of the proposed knowledge based model.

Keywords: Process Improvement, Knowledge Management, Project Management, Patterns.

1. Introduction

Software organizations are aware that having the best professionals is not everything for project success. Unless they understand the software processes of an organization, these professionals cannot perform productive and high quality software projects [1].

One of the research fields which goal is the improvement of the quality and productivity of software systems is the one based on software improvement approaches. In this field the two main reference models are the one proposed by the Software Engineering Institute named CMMI [2] and the one defined by ISO [3].

The implementation of a software process improvement program is very expensive, especially for SMEs (Small and Medium Enterprises), and those organizations that undertake firstly an initiative of this type [4].

Some other important factors that make difficult the appropriate implementation of process improvement programs are the next ones:

- Current reference models do not provide a detailed process definition, and the organization needs a great deal of resources and time to define, in an integrated way, the processes mentioned in the standards and the reference models.
- There are software tools that support process improvement programs, but due to the cost of these tools most software organizations do not have the technology and the appropriate tools to evaluate and to define their processes. Also the most important weaknesses of these tools are:
  - The software process evaluation tools are not connected to process definition tools.
  - Current tools do not gather organization's knowledge.
  - Current tools do not allow information gathering to carry out improvement plans.
  - The gathering and classification of instances of the organization's assets, and their instantiation into project's assets.

Based on the importance for organizations to develop improvement programmes to be competitive, and based on the handicaps organizations have to implement this programmes due to the cost and time these programmes require, it is necessary to make improvement programmes accessible for the most part of organizations independently of each organization features.

Based on the experience of the Federal Aviation Administration (FAA) of the United States of America, knowledge management combined with process improvement results in a positive interaction that benefits the organization and the process improvement programmes [1].

Knowledge management is based on four elements which are *data-information-knowledge-innovation* and by embracing the concept of Knowledge management and process improvement simultaneously; the FAA is experiencing a cultural change as illustrates Fig. 1 [1]. Some other works like the one published in [5], suggests that a knowledge infrastructure consisting of technology, structure and culture along with knowledge process architecture of acquisition, conversion, application, and protection are essential organizational capabilities or "preconditions" for effective

knowledge management.

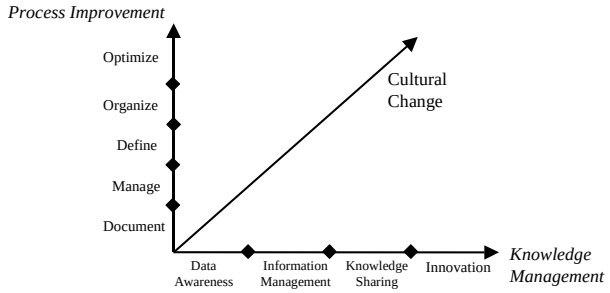

Fig. 1 FAA cultural change results

We believe that these cultural changes have many implications related with changes in the definition of process models and in the way this models must be implemented. In this sense knowledge management discipline can work together with the software engineering one, in order to translate software engineering data and information described as process models, standards, methodologies, etc., to knowledge and innovation once the knowledge of experts on process model, standards, methodologies, etc., is elicited, and translated into a computable model, so a software system can deal with this knowledge in order to reduce the cost in the process definition, and increase the maturity of the processes faster.

Process improvement can be applied in a very wide range of processes. However it is widely recognised the importance of project management in the failure or success of a project [6] [7] [8]. This is why we are going to focus this work on the improvement of a specific set of processes which are project management ones.

The research results included in this paper are divided into two main phases;

- the first one which represent the previous work of this research team to demonstrate the importance and the validity of working together in the fields of knowledge management and software process improvement field. These research results are illustrated in the paper through a brief description of the PIBOK-Model and the PIBOK-Tool, as well as the results obtained from the use of both.

- the second one, which improved the first one and represents the current work, using the concept Product patterns.

This paper is structured as follow; section 2 describes the PIBOK-Model in a static way; focusing on the model components and in a dynamic way explaining the way the model should be used. Section 3 summarises the results obtained using the PIBOK-model in real projects. Section 4 presents the main conclusions obtained and section 5 is dedicated to describe the improvements under development to the explained version of PIBOK-Model.

2. Knowledge based model approach to process improvement

As we explained in the previous section, we believe as well as the above mentioned authors, that a knowledge based approach can enhance the implementation of improvement programmes. This section is dedicated to describe the results obtained from the use of the PIBOK-Tool supporting the PIBOK-Model, as well as the improvements to the above mentioned model and tool after the analysis of the data obtained from organizations.

With the description of the problem and our hypothesis, the main goals of the work achieved since 2002 to 2004, was:

1. To develop a knowledge based software process improvement model (PIBOK-Model: Process Improvement Based On Knowledge management Model) and a support tool (PIBOK-Tool: Process Improvement Based On Knowledge management Tool) that would allow organizations to evaluate the current state of their processes, and assist in defining their project management processes.
2. To determine the validity and ease of use of PIBOK-Tool in assessing and defining the organization's project management processes.

In order to provide the infrastructure needed to support the proposed PIBOK-Model a set of components has been identified and the way the PIBOK-model must be used has been defined. Fig. 2 summarizes both, the components and the procedure to use the PIBOK-Model.

PIBOK-Model is intended with the aim to improve software project management processes based on the standard PMBOK (Project Management Body of Knowledge) [9], software engineering reference models such as SW-CMM [10], CMMI [2][11], ISO 15504 [3], etc. and the most important project management methodologies such as Prince2 and Métrica3, DOIT, TenStep, etc. Fig. 2 shows the logical architecture of the model.

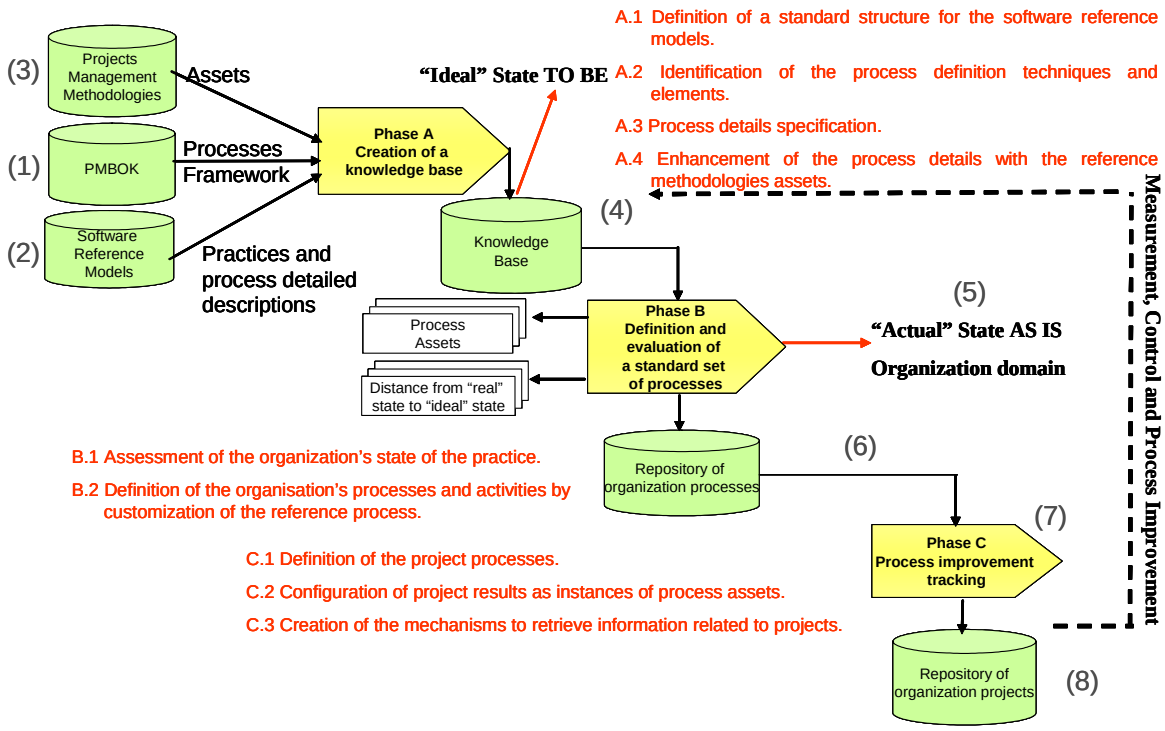

Fig. 2 Architecture of the PIBOK-Model

Each component of the model is identified with a number in brackets, next appears the description of each component:

1. The PMBOK process framework configures the core of PIBOK-Model.
2. The PMBOK processes are detailed using experts opinion and based on the software reference models practices, for example, SW-CMM, CMMI, ISO 15504, etc.
3. The process details are enhanced with the process assets of the most important project management methodologies for example Métrica 3, Prince2, TenStep, etc.
4. The creation of a knowledge base that contains the meta software project management process definitions is the result of this model.
5. Using PIBOK-Model, all the organization's process assets are gathered during the process assessment phase. These process assets enhance the organization's software project management process definition. The software organization's process definitions and its assessment information are stored in the knowledge base.
6. As an aid to improving the processes, PIBOK-Model also offers the possibility to adapt generic process assets, which are stored in the knowledge base and come from the most important project management methodologies. The standard software project management processes will be defined from the assessment results in a semi-automatic way.
7. Once the organization has the standard definition of its software project management processes, the PIBOK-Model allows adapting the organization's standard processes to each concrete project.
8. The knowledge base also allows storing the products generated during the execution of management activities configuring an historical data base of software project management within the organization.

3. Experimentation and results obtained

The PIBOK-model is supported by a web based tool called PIBOK-Tool that has been evaluated in some organizations.

PIBOK (Model and Tool) has been experimented on several software organizations

by training high-level managers in the model concepts and its use in several software process improvement programs. Later, several actions of verification have been made, that are:

- 8 high-level managers and decision makers, who determine the strategy of the organization and are committed to carrying out an improvement programs, were interviewed to determine the capacity of PIBOK-Model to define the improvement objectives and track the evolution of the improvement

- 11 team leaders responsible for improvement programs execution or experts in software process improvement projects, who had used the PIBOK-Tool, were interviewed to determine the capacity of PIBOK-Model to provide efficient mechanisms for managing software process improvement projects.

- Finally, 32 people in charge to define new software processes were observed while they are using PIBOK-Tool for this purpose in order to determine the capacity of PIBOK-Model to design software processes. The results obtained were compared with control information from other improvement programs involving 33 people

4. Conclusions

As main conclusion, these experimentation activities provide evidences to determine that PIBOK-Model allows software organizations to gather classify and manage the organization's knowledge (its active process assets). It also allows organizations to adapt their own process assets, thus providing:

- Alternatives (software management process assets methodologies).

- The PIBOK-Model offers the organization a thesaurus of software project management for product reuse, standards, methodologies, assets, etc.

- As a result of the processes assessment, the PIBOK-Model offers software organizations two process definitions (each with its own process assets) and the state of the real current process definition. Having two software process definitions helps the organization to know the strengths and weaknesses at all times and to establish improvement priorities.

- The PIBOK-Model acts as a historical knowledge base of the practices used by the organization to manage software projects.

- The PIBOK-Model permits consultation and analysis of the different instances of the process assets stored in the historical data base. The analysis of the information facilitates the detection of the strengths and weaknesses of the organization's software process.

5. Future trends and PIBOK improvements

Despite of the good results obtained using the PIBOK-Model during experimentation phase, there are some parameters we would like to focus on in an improved version. If we take a look at the obtained results using PIBOK-Model and Tool, we realize that the efficiency of use and knowledge management capacity should be improved.

For achieving these purposes, we are working on:

- the improvement of the efficiency of use of the PIBOK-Model (understood as the number of tasks performed by time unit), moving from the actual ETVX (entry tasks verification exit) format to a graphic representation.

- the achievement of the next two knowledge management stages, knowledge and innovation.

The concrete changes, which are already approved and in course, are described in detail below.

5.1. Changes to process definition technique

The first change adopted is related to the way to represent the definition of software processes. Currently, the processes are defined using an extended ETVX definition technique, containing the following information items: Purpose, Preceding Processes/Activities, Subsequent Processes /Activities, Entry Criteria, Inputs, Activities / Tasks, Outputs, Exit Criteria,

Practices, Tools and techniques, Metrics/Measurements, Interfaces with other processes, Roles and Notes.

The new way to define the processes is SPEM [13]. SPEM is a conceptual model to define processes based on UML extensions, providing a formal language that is a XMI extension and a graphical language to represent processes following UML basic diagrams.

In order to provide the functionalities related to process definition using SPEM, we are working on several adaptations related to:

- The modification of internal structure of PI-BOK repository to maintain specific information of processes considered in SPEM but not in extended ETVX.

- The adaptation of SPEM to PIBOK-Model needs. To satisfy all our requirements, SPEM model has to be extended to include information related to practices, lessons learned, metrics and measurements information.

- The modification of PIBOK-Tool user interface, because currently the way to define a software process is exclusively based text fields. SPEM provides a graphic language to represent the concepts to model software process, so the new user interface will be based on graphic components to draw the processes.

5.2. Changes to process definition technique

In the last version of PIBOK-Model, the software processes are defined as a set of activities that are extracted from PM-BOK and enriched with information coming from software process reference models and the most outstanding software development methods.

In order to improve the knowledge management capacity of PIBOK-Model, the software process will be now defined in terms of the products to be elaborated and used during the process execution. All the products available to define the processes are also extracted from PM-BOK and enriched with information from the most outstanding software development methods. To determine how the products should be elaborated, updated and used during a software process, we have defined a new concept called "product pattern".

The concept *"product pattern"* is a new term that comes from the Alexandrian Patterns [12], and is intended to gather the knowledge of software engineering experts to obtain a specific software product, understanding product as any thing to be produced during the whole software development process. This product pattern is described in terms of the next fields:

- **Name**: name of the product pattern
- **Related patterns**
- **Initial Context:** Present situation where the project is being executed.
- **Resulting Context:** Future situation as a result of executing the pattern.
- **Problem:** Improvements to be achieved.
- **Forces:** Forces can come from different sources. We have identified the next sources of forces:
  - Organization features
  - Kind of system to be developed
  - Kind of client
  - Market Scope
- **Solution**: instantiations of products previously obtained in this context with this problem entailing these forces. Including also time for completion.
- **Entries**: previous obtained products necessary to develop this one.

The definition of the product patterns solution and entries will be done using SPEM, but the rest of information (Initial context, resulting context, problem description and forces) will be implemented using conceptual models.

These conceptual models are used because they allow the use of knowledge management recovery techniques and transformation of the experts knowledge in innovation. This transformation affects to the component (4) in Fig. 2, which in the next version of PIBOK-Model, from now PIBOK-PB-Model, is represented with *product patterns* instead of with ETVX format.

In the new version of PIBOK-PB-Model, the selection of the concept *product pattern* as the element to encapsulate the knowledge is based on the idea that a product is the minimum software engineering element to be obtained in any process model execution and the same product can be involved in different process model. This is why we believe that a "product" knowledge based solution is more flexible and reusable than knowledge based process solution.

In order to execute a software project, the project manager will find the appropriate *product patterns* from (4) in Fig. 2, following the next

rule:

If you find yourself
in this context,
with this problem
entailing these forces
then
map a product pattern in your project
look for product patterns

Actually we are working with software engineering experts in the identification of the criteria that allow the appropriate classification and recovery of product patterns according to the context, problem and forces of the project to be developed in which the product patterns are being used.

6. Acknowledgements

This work has been partially funded by the Spanish Ministry of Science and Technology through project TIC2004-7083.

7. References

[1] Burke, D., Howard, W. Knowledge Management and Process Improvement: A Union of Two Disciplines. The journal of defense software engineering. Jun 2005. Available at: http://www.stsc.hill.af.mil/crosstalk/2005/06/0506Burke.html.

[2] Software Engineering Institute. "CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI-SE/SW/IPPD/SS, V1.1)", Carnegie Mellon University, March, 2002.
Standard ISO/IEC 15504-5:1999 Standard for Information Technology-Software process assessment. 1999.

[3] Standard ISO/IEC 15504-5:1999 Standard for Information Technology-Software process assessment. 1999.

[4] Pinto R. and Shoemaker D. “The Cost of CMM in a Conventional IT Organization: A Field Study”, 2002.

[5] Gold, A., Malhotra, A., and Segars, A. Knowledge Management: An Organizational Capabilities Perspective. Journal of Management Information Systems. (2001). Vol. 18 No. 1, pp. 185 – 214.

[6] White, D., Fortune, J., Current practice in project management – an empirical study. (2002). International Journal of Project Management. Vol. 20, Issue 1, Pp. 1-11.

[7] Nienaber, R., Cloete, E. A software agent framework for te support of software project management. (2003). Proceedings of the SAICSIT 2003. Pp 16-23.

[8] McConnell, E., Nine Deadly Sins of Project Management. From the Editor IEEE Software, September/October 2001. Available at: http://www.stevemcconnell.com/ieeesoftware/iec19.htm.

[9] Project Management Institute (PMI), “The Project Management Body of Knowledge (PMBOK)”, Project Management Institute, Upper Darby, PA, 1987.

[10] Paulk M., Garcia S., Chrissis M., and Bush M. Capability Maturity Model for Software, Version 1.1, CMU/SEI-93-TR-24. Technical Report. Software Engineering Institute. Carnegie Mellon University, 1993.

[11] Phillips M. “CMMI V1.1 Tutorial”. Software Engineering Institute. Carnegie Mellon University, 2003.

[12] Alexander, C. A Pattern Language : Towns, Buildings, Construction. Oxford University Pres. (1977).

[13] OMG, Software Process Engineering Metamodel Specification (SPEM) Versión 1.1, January 2005.