# A COLLABORATIVE FRAMEWORK TO SUPPORT SOFTWARE PROCESS IMPROVEMENT BASED ON THE REUSE OF PROCESS ASSETS

Fuensanta Medina-Domínguez, Javier Saldaña-Ramos, Arturo Mora-Soto
Ana Sanz-Esteban and Maria-Isabel Sanchez-Segura
*Computer Science Department, Carlos III Technical University of Madrid*
*Avda. Universidad, 30, Leganes 28911, Madrid, Spain*

Keywords: Software Process Improvement, Process Assets, Product Pattern, Patterns, Knowledge Management, Reuse, Software Engineering.

Abstract: In order to allow software organizations to reuse their know-how, the authors have defined product patterns artefact. This know-how can be used in combination with software engineering best practices to improve the quality and productivity of their software projects, as well as reduce projects cost. This paper describes the structure of the Process Assets Library (PAL), and the framework developed to encapsulate the know-how in organizations. The PIBOK-PB (Process improvement based on knowledge - pattern based) tool uses the proposed PAL to access the knowledge encapsulated in the product patterns, and to execute software projects more efficiently. This paper also describes PIBOK-PB's features and compares similar tools in the market.

## 1 INTRODUCTION

In spite of the fact that the use of software process best practices in a software project development improves productivity in organizations, projects planning and the quality of the software products (Withers, 2000), very few organizations implement these best practices (Richard, 2003) (Soumitra Dutta, 1999). Currently, software process best practices are not deployed because of the little usability of project management tools. They present several constraints to represent, manage and transfer the knowledge of these best practices and the knowledge obtained through the software engineering experts' experience.

Knowledge management is defined as "the set of systematic and disciplined actions that an organization can take to obtain the greatest value from the knowledge available to it" (Marwick, 2001). Knowledge management is based on four elements (Hey, 2004): *data* is the minimum unit of information not elaborated; *information* is a set of inter-related data to acquire sense*;* knowledge is defined as expertise, and skills acquired by a person through experience or education; *innovation* is the successful exploitation of new ideas. There are two kinds of knowledge: explicit and tacit. Explicit knowledge can be codified into documents and databases and shared among stakeholders relatively easily. Tacit knowledge is the know-how of organizations, the capabilities and experiences of their stakeholders.

This knowledge is difficult to formalize, communicate and share among the people involved in the project in the organization.

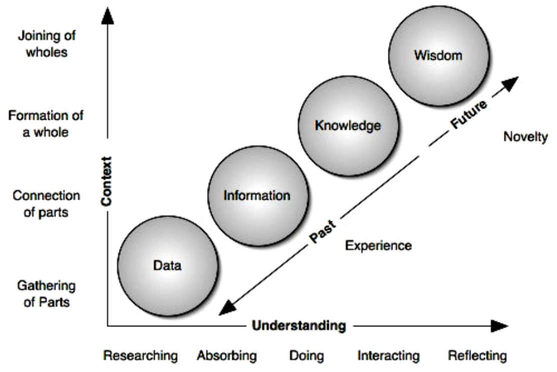


Figure 1: One view of the DIKW hierarchy. (Clark, 2004).

This tacit knowledge is essential to achieve the innovation stage so that the organization will gain the competitive advantage.

Software engineers look for knowledge sources, for example books, on-line resources, databases, to develop a software process but, on many occasions, this knowledge is the tacit knowledge of the organizations (Assimakopoulos, 2006). The study carried out by (Komi-Sirviö et al, 2002) revealed that the knowledge is not easy to find and, when it is discovered, it cannot be reused. The reasons are that knowledge gathering is too informal and the knowledge is not readily available to the organizations. Consequently, it is necessary to gather the know-how of the organizations in an artefact to be retrieved and reused in subsequent projects. We defined an artefact, called product patterns (Amescua et al, 2006) (Medina-Dominguez, 2007), to gather the knowledge of the software engineering experts to produce a software product.

In this paper we describe the collaborative framework that supports software process improvement based on process assets reuse. The main capabilities of this framework are:

- to reuse and manage of process assets
- to improve the efficiency of use of the processes
- to reduce costs in software process improvement programs
- to work collaboratively in the phases of software projects

The remainder of this paper is structured as follows: section 2 compares the proposed framework with similar ones in the market. Section 3 explains the collaboration framework proposed for process assets reuse in the project under development. And, finally, in section 4 we present the conclusions and future works.

## 2 RELATED WORKS

The purpose of this paper is to propose the most accurate architecture for the collaborative framework. So, the authors reviewed some tools available in the market as well as some research prototypes to find out the most critical gaps in the software tools that software engineers use to manage software process improvement programs and projects execution.

We reviewed four software tools and three research prototypes. The authors analysed, and evaluated the selected tools based on sixteen features rated: 0 (Not present), 1 (Partially present), 2 (Present). The features evaluated were inspired by the work presented by (Dargan, 2001) and are presented together with the results in http://sel.inf.uc3m.es/projects/PIBOK-PB/AnalysisRelatedWorks.pdf. In the reminder of this section we describe the main results of the evaluation.

### 2.1 Support Software Tools

The software tools analysed are presented, mainly focusing on the following features: knowledge management, know-how reuse, project management, software process models deployment support, collaborative development platform and knowledge searching.

- CodeBeamer (Intland Software, 2007): is a collaborative development platform based on J2EE that offers application life cycle management features for development teams. It includes the following main capabilities: knowledge management (through a mechanism called wiki asset linking), project management and information retrieval. This tool lacks knowledge management formalism because wikis are considered content management systems (Cunningham, 2001) instead of a formal knowledge management artefact. This explains why knowledge reuse cannot not be done automatically. Despite the fact that this tool is intended to support software development, it does not provide either a mechanism to select a software process model for the organization and project features or an electronic process guide to perform project activities.

- IRIS Process Author (Osellus, 2007): is a visual process management system that enables collaborative authoring and tailoring of process assets. This tool focuses on process management and, for project management, offers the possibility of exporting process content and configuration to third party tools such as Microsoft Project. Although this application also includes some knowledge management features, it lacks knowledge reuse for the organization and projects context and constraints. For know-how representation it uses templates and wikis instead of a formal representation artefact such as patterns, ontologies or a thesaurus. Another deficiency is the lack of software process models management, project planning and project tracking.

- Microsoft Visual Studio Team System (VSTS) (Microsoft Corporation. 2007): is an integrated Application Life-Cycle Management (ALM) solution comprising tools, processes, and guidance to help everyone on a development team to improve their skills and work more and more effectively together. As mentioned in (Medina-Dominguez, 2007), VSTS presents some deficiencies. Therefore, we decided to develop a collaborative framework to cover the deficiencies related to knowledge management, knowledge reuse and project management.

- Select Solution Factory (Select Business Solutions, 2006) this tool connects component based development and solution assembly. It allows organizations to manage complex software development projects as well as to implement code and software components reuse. Although this application is intended to ease software development and deployment, knowledge management features are absent in reuse. This tool offers support for some modelling methods such as SSADM and Yourdon but lacks software process models management support.

## 2.2 Research Prototypes

This section presents some prototypes, developed through research works, which were analysed according to the following features: knowledge management, know-how reuse, project management, software process models deployment support, collaborative development platform and knowledge tracking.

- BORE (Henninger, 2003) is a prototype tool designed to further explore and refine the requirements for tools supporting experience-based approaches. This tool is aimed at reusing organizational experience by packaging it in experience repositories. Although this tool offers a huge functionality for knowledge management, its main flaw is project management. In spite of presenting tasks definitions for team members roles, it lacks vision of the project as a whole.

- OnSSPKR (He, 2007): is a knowledge management based tool for supporting software process improvement. It is aimed at composing and organizing useful process assets into a knowledge repository based on ontology. It also supports software process knowledge storage for retrieval. This tool offers neither project management execution functionalities nor software process models management. It does not support a clear collaborative framework in spite having of web portals implemented.

- ProKnowHow (Silveira Borges et al, 2002): is a knowledge management based tool for supporting software process improvement. It contains formal and informal knowledge in the software process database of an organization. It uses two ontologies to ground the structure of the organization's memory. This tools offers project management and some search capabilities for accessing past process plans and lessons learned. It lacks practical representation for knowledge reuse. Neither active electronic software process guide nor a collaborative environment for project execution and management is offered.

## 3 COLLABORATIVE FRAMEWORK DEFINITION

PIBOK-PB is a collaborative framework to support software process improvement and is based on process assets reuse. With this tool the authors tried to fill all the gaps identified in the tools selected and explained in Section 2.

The PIBOK-PB architecture is shown in Figure 2. The following sections describe the PIBOK-PB framework in more detail.

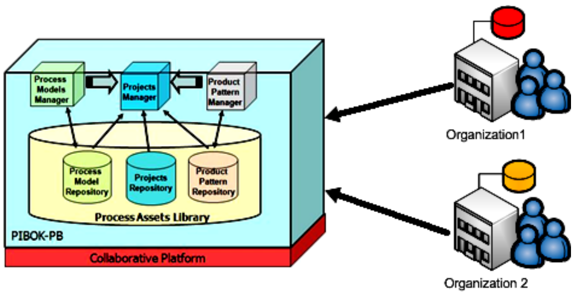


Figure 2: PIBOK-PB Architecture.

## 3.1 PIBOK-PB Main Features

The main functionalities of the PIBOK-PB tool were described in (Medina-Dominguez, 2007). The most relevant are: 1) the reuse of process assets supported by product patterns artefacts (Amescua et al, 2006) (Medina-Dominguez, 2007); 2) knowledge management capabilities; and 3) software process models selection for the organization and project features.

The collaboration layer was improved in the PIBOK-PB tool by developing a collaborative framework to fill the gaps in knowledge management, knowledge reuse and project management. The authors believe that this collaborative framework will facilitate the use of PIBOK-PB in organizations, and collaboration among those organizations that decide to share knowledge and information.

## 3.2 Knowledge Managements Components

The collaborative framework is made up of three knowledge management components as follows.

**Software Process Model Manager.** This component provides several software process models for each project under development. The project manager decides which model best fits the information provided by this component for the organization and project features. In addition, the project manager can create a specific model process. The Software Process Model Manager offers a set of methodologies or activities to construct different processes such as software analysis, software design, software management, software development and software deployment processes. In this way the organizations can customize their own software processes, creating a specific software process model.

**Product Pattern Manager.** This component provides the knowledge that a stakeholder needs to perform each process model activity. In order to achieve this task, the system looks for the product pattern that best fits each activity. In this search, the component takes into account the context, project and constraints of the organization, and the problem to be solved. The rule used to select the accurate pattern is:

```
If you find yourself in this context
  (and) with this problem
  (and) entailing these forces
then
  map a product pattern in your project
  (and) look for more product patterns
```

All product patterns that best fit the input variables are shown to the project manager. One of the most important fields of product patterns is the solution. This field provides the steps to carry out this activity and obtain a specific software product. The project manager has to select and instantiate a product pattern for each activity based on his/her experience.

**Current Project Manager.** This component provides the execution plan for the current software project through an active electronic process guide and an electronic execution project guide. The active electronic process guide provides an overview of the software process model with the following elements: a software project planning, the tasks of each team member role and a workflow representing the processes of the project. The electronic execution project guide provides: 1) the product patterns showing the steps to obtain the defined software product; 2) templates where stakeholders can capture the information on the current project; 3) lessons learned by experts engineers; and 4) relevant information on each activity.

## 3.3 PIBOK-PB Process Assets Library Description

A process asset has no value if it is not easily accessible. The process assets have to be organized, well-indexed and easily accessible to stakeholders who need process guidance information. These characteristics are provided by a Process Assets Library (PAL).

The PIBOK-PB Process Assets Library is a collaborative knowledge base and a central repository that is made up of three repositories:

**Process Model Repository.** this repository stores the software process best practices, including software process models[1], methodologies[2] and techniques applied to software processes. The knowledge stored contains the activities and tasks of these software process best practices, roles of each activity and relevant information of each., This repository also contains a set of heuristics based on rules which, for a set of features of an organization and a set of project features to be developed, provides the process model, methodology and techniques that best fit these features.

[1] Capability Maturity Model Integration –CMMI- (CMMI, 2001), ISO/IEC 15504, also known as SPICE -Software Process Improvement and Capability dEtermination- (ISO/IEC 15504, 2005), etc.

[2] Unified Software Development Process –UP- (Jacobson, 1999), extreme Programming –XP- (Beck, 2002), Personal Software Process –PSP- (Humphrey, 2005), Team Software Process –TSP- (Humphrey, 2005), (Humphrey, 2006), Scrum Development Process, etc.

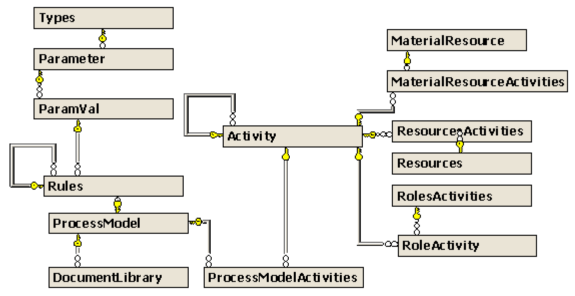


Figure 3: Process Model Repository.

**Product Patterns Repository.** this repository stores the knowledge of software engineering experts in order to obtain a specific software product and the know-how of the organization. All this knowledge provides stakeholders with the information needed to create, develop and deploy new and better processes so that the organization achieves the innovation stage of knowledge management

. This repository explains how a specific role can perform a specific activity.

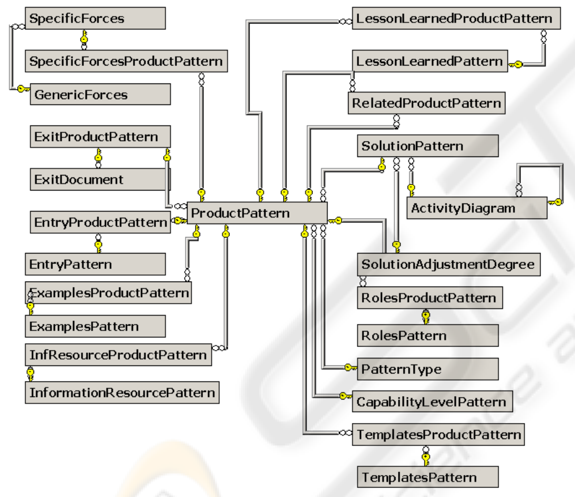


Figure 4: Product Patterns Repository.

**Project Repository.** this repository stores software projects that have already been developed as well as those under development. It contains three types of information related to:

- the process model chosen for this project: process model, activities, roles, workflow;
- the product patterns of each process model activity which provide enough information to perform each activity, and the active electronic process guide where this information is shown to the stakeholders;
- the execution and development of each activity where the information stored is: project planning, products obtained from each activity and project tracking.

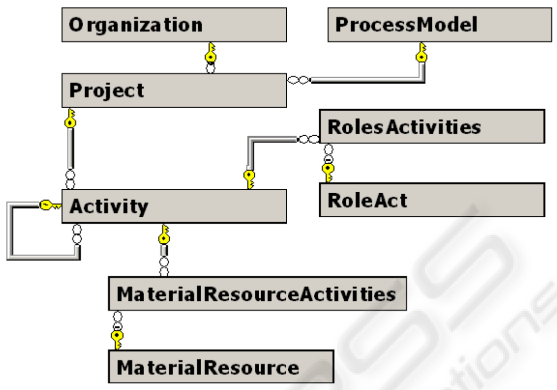


Figure 5: Project Repository.

### 3.4 PIBOK-PB Collaborative Support

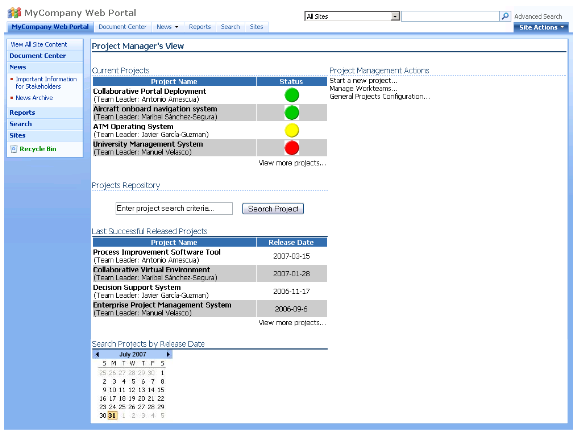
The organizations need to share knowledge among stakeholders to enable them to develop the project activities, taking advantage of the knowledge of past projects and the experience and knowledge of the software engineering experts. In this way, stakeholders can perform their daily tasks and make decisions using the maximum information available. So, organizations have to commit themselves to providing the mechanisms and technologies to allow collaboration among stakeholders, coordinating activities and sharing knowledge to develop software project processes without increasing the time and cost of projects. In order to disseminate this knowledge, the framework is supported by a collaborative platform.

The collaborative platform proposed by the authors is based on the Microsoft Office SharePoint Portal Server (MOSS) because of the following capabilities:

- Better communication among different software projects roles.
- Coordination of the activities among teamwork members who need synchronous and/or asynchronous interaction regardless of their geographical location.
- Integration among different systems in the organization.
- Extension mechanisms through application programming interfaces.
- The most advanced search engine with many superior characteristics over its competitors

such as meta data management or automatic language detection.
- A familiar and user-friendly interface.

Two screenshots of the collaborative framework prototype developed are shown below. This prototype is available at http:// sel.inf.uc3m.es/demos/synergy



Project Manager's View

Current Projects

| Project Name | Status |
|---|---|
| Collaborative Portal Deployment (Team Leader: Antonio Amescua) | 🟢 |
| Aircraft onboard navigation system (Team Leader: Maribel Sánchez-Segura) | 🟢 |
| ATM Operating System (Team Leader: Javier García-Guzman) | 🟡 |
| University Management System (Team Leader: Manuel Velasco) | 🔴 |

View more projects...

Project Management Actions

Start a new project...
Manage Workteams...
General Projects Configuration...

Projects Repository

Last Successful Released Projects

| Project Name | Release Date |
|---|---|
| Process Improvement Software Tool (Team Leader: Antonio Amescua) | 2007-03-15 |
| Collaborative Virtual Environment (Team Leader: Maribel Sánchez-Segura) | 2007-01-28 |
| Decision Support System (Team Leader: Javier García-Guzman) | 2006-11-17 |
| Enterprise Project Management System (Team Leader: Manuel Velasco) | 2006-09-6 |

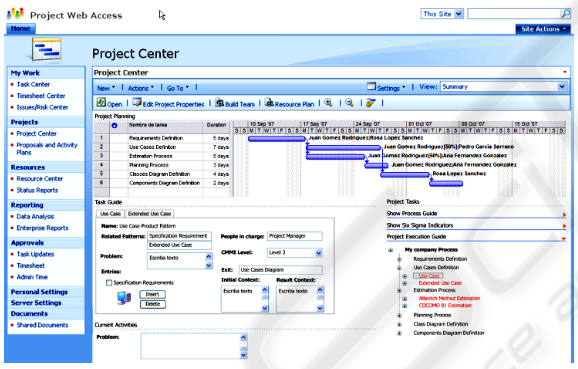
View more projects...

Search Projects by Release Date

July 2007

| S | M | T | W | T | F | S |
|---|---|---|---|---|---|---|
| 25 | 26 | 27 | 28 | 29 | 30 | 1 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | 1 | 2 | 3 | 4 | 5 |

Figure 6: Project Manager's Dashboard View.



Project Center

Project Center

View: Summary

Project Planning

|  | Nombre de tarea | Duration | Resources |
|---|---|---|---|
| 1 | Requirements Definition | 5 days | Juan Gomez Rodriguez;Rosa Lopez Sanchez |
| 2 | Use Cases Definiton | 7 days | Juan Gomez Rodriguez[50%];Pedro Garcia Serrano |
| 3 | Estimation Process | 5 days | Juan Gomez Rodriguez[50%];Ana Fernandez Gonzalez |
| 4 | Planning Process | 3 days | Juan Gomez Rodriguez;Ana Fernandez Gonzalez |
| 5 | Classes Diagram Definition | 4 days | Rosa Lopez Sanchez |
| 6 | Components Diagram Definition | 2 days |  |

10 Sep '07 | 17 Sep '07 | 24 Sep '07 | 01 Oct '07 | 08 Oct '07 | 15 Oct '07

Task Guide

Use Case | Extended Use Case

Name: Use Case Product Pattern
Related Patterns: Specification Requirement; Extended Use Case
People in charge: Project Manager
Problem: Escriba texto
CMMI Level: Level 3
Entries: Specification Requirements
Exit: Use Cases Diagram
Initial Context: Escriba texto
Result Context: Escriba texto

Project Tasks

My company Process
- Requirements Definition
- Use Cases Definition
  - Use Case
  - Extended Use Case
- Estimation Process
  - Albretch Method Estimation
  - COCOMO 81 Estimation
- Planning Process
- Class Diagram Definition
- Components Diagram Definition

Current Activities

Problem:

Figure 7: Project Management and Tracking Interface.

## 4 CONCLUSIONS AND FUTURE TRENDS

In this paper, we have described a collaborative framework to support software process improvement based on the reuse of process assets. The framework, called PIBOK-PB, is made up of tree knowledge management components, a process assets library and a collaborative platform. Its main goal is the transfer and reuse of knowledge among the stakeholders involved in a software project development. The knowledge stored in the PAL contains the experience of software engineering experts, the know-how of the organization and the products obtained during the execution of any software project, usually stored as non structured assets.

The key benefits and potential use of our work include:

- Selection and customization of the software process best practices that best fit the needs of the organization's features.
- Reuse of the know-how of the organization and the knowledge of the software engineering experts through product patterns artefacts.
- Enough information to know, at each stage of the software process development, what each stakeholder has to do and the steps to take through an Active Electronic Process Guide.
- Knowledge sharing and working collaboratively through the collaborative platform.
- Feedback mechanisms so that new product patterns can be created and, in this way, promote the innovation.

## ACKNOWLEDGEMENTS

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

## REFERENCES

Amescua, A. Garcia J.Segura-Sanchez, M., Medina-Dominguez F. 2006. A pattern-Based Solution to Bridge the gap between theory and practice in Using process models. In LNCS, vol. 3966, pp. 97-104. Springer.
Assimakopoulos,D., Yan J. 2006. Sources ofknowledge acquisition for Chinese software engineers. R&D Mamagement .
Beck, K. 2002. eXtreme Programming Explained. Addison-Wesley.
Clark D. 2004. The Continuum of Understanding. http://www.nwlink.com/~donclark/performance/understanding.html
CMMI SM Product Suite. 2001. Retrieved from www.sei.cmu.edu/cmmi/products/products.htm
Cunningham, W., Leuf, B. 2001. The Wiki Way. Addison-Wesley Professional.
Dargan, P. A. 2001. The Ideal Collaborative Environment. The Journal of Defense Software Engineering.
He, J., Yan, H., Liu, C., Maozhong, J. 2007. A Framework of Ontology-Supported Knowledge Representation in Software Process.

Henninger, S. 2003.Tool Support for Experience-Based Software Development Methodologies. Advances in Computers , 59, 29-82..

Hey, J. 2004. The Data, Information, Knowledge, Wisdom Chain: The Metaphorical link

Humphrey, W.: PSP(sm). 2005. A Self-Improvement Process for Software Engineers. Addison-Wesley.

Humphrey., W. 2005. TSP(SM)-Leading a Development Team. Addison Wesley.

Humphrey, W. 2006. TSP(SM)-Coaching Development Teams. Addison Wesley.

Intland Software. 2007. codeBeamer. http:// www.intland.com/products/codebeamer.html

ISO/IEC 15504(1-5):2005. 2005 Standard for Information Technology-Software process assessment.

Jacobson, I., Booch, G., Rumbaugh, J. 1999. The Unified development process. Addison-Wesley.

Komi-Sirviö, S., Mäntyniemi, A., Seppänen, V. 2002. Knowledge Management - Toward a Practical Solution for Capturing Knowledge for Software Projects. IEEE Software; 19 (3) pp.60-62 ISSN:07407459

Marwick, A.D. 2001. Knowledge management technology. IBM System Journal. Vol 4, pp. 40

Medina-Domínguez,F. Sanchez-Segura, M., Amescua, A, Garcia, J. 2007. Extending Microsoft Team Foundation Server Architecture to support Collaborative Product Patterns. LNCS 1-11.

Microsoft Corporation. 2007. Visual Studio Team System. http://msdn2.microsoft.com/en-us/teamsystem/default.aspxOsellus. 2007. IRIS Process Author. http://www.osellus.com/IRIS-PA

Select Business Solutions. 2006. Select Solution Factory. from http://www.selectbs.com/products/select-solution-factory.htmSoumitra Dutta, M. L. 1999. Software Engineering in Europe: A study of Best Practices

Silveira Borges, L. d., Almeida Falbo, R. 2002. Managing Software Process Knowledge.

Richard, T. 2003. Seven Pitfalls to Avoid in the Hunt for Best Practices. 67-69.

Withers, D. 2000. Software engineering best practices applied to the modeling process. 432-439.