RAMALA: A KNOWLEDGE BASE FOR SOFTWARE PROCESS IMPROVEMENT

Y. Rimawi

Computer Science Department, Carlos III University of Madrid, Avda. de la Universidad 30, 28911 Leganes, Madrid, Spain

A. Amescua

Computer Science Department, Carlos III University of Madrid, Avda. de la Universidad 30, 28911 Leganes, Madrid, Spain

G. Cuevas

Computer Science Department, Polytechnic University of Madrid, Facultad de Informática, 28660 Boadilla del Monte, Madrid, Spain

T. San Feliu

Computer Science Department, Polytechnic University of Madrid, Facultad de Informática, 28660 Boadilla del Monte, Madrid, Spain

J. Garcia

Computer Science Department, Carlos III University of Madrid, Avda. de la Universidad 30, 28911 Leganes, Madrid, Spain

Correspondence Email: yrimawi@gmail.com

ABSTRACT

Software process engineering has not been introduced in most small software organizations because of the high cost of implementing it. RAMALA is a knowledge base, supported by a software tool which is also called RAMALA. This knowledge base contains a software process framework based mainly on PMBOK [9], using the best practices of the main software process reference models such as CMMI [11] and ISO 15504 [3], and enriched with process assets of the most outstanding software process development methodologies. RAMALA is a platform in which the best practices are collected. As a result, RAMALA makes it possible for small software organizations to define, assess and improve their software processes easily and economically.

Keywords: Software Process Reengineering, Process Modeling, Software Process Improvement, and Software Engineering.

1. INTRODUCTION

Software production in most small software organizations is characterized by poor management and skills, typical features of the 'Software Crisis'. These give rise to serious problems such as project delay, high costs, and poor quality products.

The software community has been trying to deal with this crisis for almost four decades. In 1968, Nato held a conference in which the term "Software Engineering" was coined [7]. Ever since, Software Engineering has been a growing discipline, enriched by two basic movements: technological and process. While reprogramming languages, software tools and techniques characterize the technological movement, the process movement is characterized by reference models, software process improvement elements and their support activities.

During the last decade, the process movement expanded widely in numerous organizations where it was found that the major software problems were due to inefficient management of the software process. One of the reports of the Department of Defense of the United States of America [1] states: "*After two decades of largely unfulfilled promises about productivity and quality gains from applying new software methodologies and technologies, industry and government organizations are realizing that their fundamental problem is the inability to manage the software process*".

Several organizations have contributed to the process movement through the development of software process reference models and standards such as CMM [5] [6], CMMI [11], ISO 15504 [3], and PMBOK [9]. Although there are several software process reference models and standards that software organizations can select from to improve their software processes, few organizations implement any due, among other reasons, to the high cost of a software process improvement program. For example, from April 2002 through December 2004, just 567 organizations around the world had conducted SCAMPI v1.1 class A appraisal [12].

The SEI carried out a study in response to a demand for information on the results of software process improvement efforts [4]. This study covered 13 organizations that represent a variety of maturity levels. The results showed that the average yearly cost of software process improvement was $245,000 and the average number of years taken was 3.5. This means that implementing a software process improvement program is very expensive, especially for small and medium-sized companies.

The results of another study carried out to calculate the cost of CMM deployment by activities in a conventional IT organization [8] is shown in Table 1.

Table 1: Cost of CMM deployment activities

Activity Category	Percent
CMM Process Flow Specification	19.90%
CMM Control Flow Specification	13.92%
CMM Data Flow Specification	11.53%
Decision Maker Management	26.70%
Product Related Process Assurance Activity	22.29%
Initial Training	3.48%
On-going training	2.18%

As we can see from the table, the first three activities (in red) are related to process definition and make up more than 45% of the total cost; the process assurance activity (also in red) is more than 22%. These four activities alone make up 67.64% of the total cost.

This means that software engineering experts account for most of the cost of deploying the software process reference model. We believe that this percentage can be reduced by using a knowledge base supported by a software tool which gathers and manages the experts' knowledge.

(RAMALA gathers the software engineering knowledge needed to deploy a software process improvement program in a software organization.) RAMALA meets three main requirements in a software process improvement program: process assessment, process definition, and process improvement tracking.

2. RAMALA KNOWLEDGE BASE

RAMALA knowledge base is the result of a research in the Computer Science Department at Carlos III University of Madrid [13]. Its main scope and goal was to model and develop a knowledge base for software process improvement, supported by a software tool that enables the definition, assessment, and improvement of the software processes of an organization.

This knowledge base structure is shown in Figure 1. As we can see, the process definition functionality is covered by the process improvement component, with the PMBOK Guide Process Framework [9] as its core.

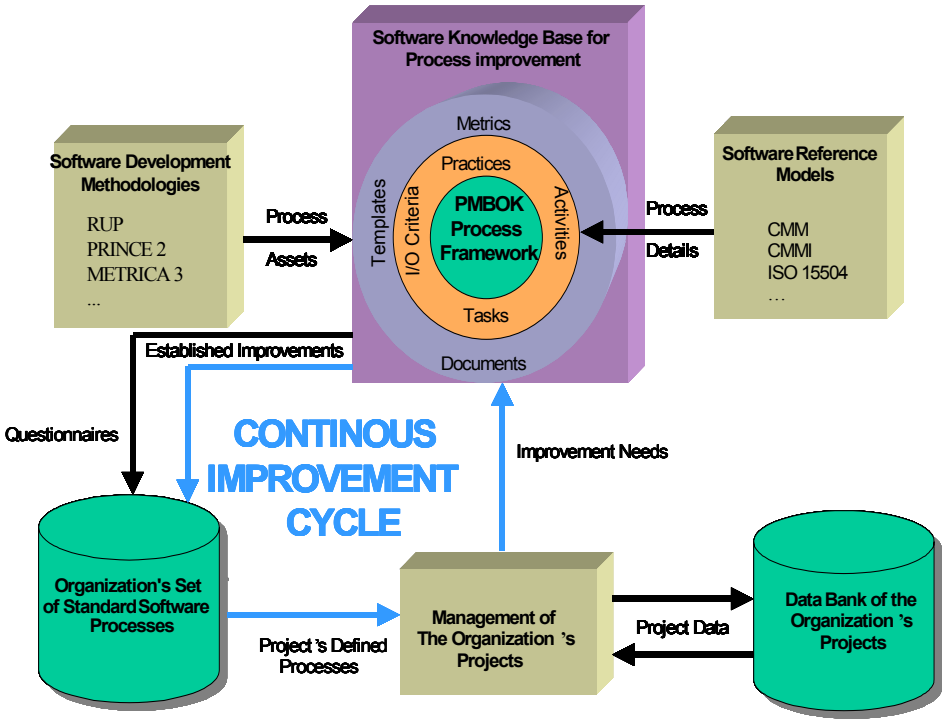

Figure 1: RAMALA knowledge base structure

Software engineering experts, using the best practices of the software process reference models and process assets of the most outstanding software process development methodologies, detail the process framework.

A formal assessment method, valid for any software process reference model, covers the process assessment functionality. During the assessment phase, RAMALA gathers and classifies all the process assets in the organization and links them to the related software process elements. Along with the assessment result, which is a color snapshot of the knowledge base, RAMALA provides the set of standard software processes of the organization.

A mechanism in the improvement functionality establishes the project's defined processes, manages the project's process assets instances, and gathers measure data to verify the fulfillment of the improvements.

The RAMALA knowledge base is described in detail below.

2.1 Software process engineering knowledge base for software process improvement

In order to build a standard and robust software process engineering knowledge base for software process improvement, the following requirements have to be satisfied:

1. Standard structure for software process reference models
2. Standard process framework
3. Formal process definition

Standard structure for software process reference models

RAMALA uses a generic data model which is able to save all the elements of each software process reference model such as CMM [5] [6], CMMI [11], and ISO 15504 [3] in one repository.

Standard process framework

Although software process reference models help organizations to define their software processes, organizations need a standard framework to define these processes in an integrated way.

The Project Management Institute (PMI) has developed an international project management standard: the Project Management Body of Knowledge (PMBOK) Guide [9]. This standard provides a process framework where all the processes, and their dependencies, necessary to manage any project are identified. RAMALA uses the PMBOK Guide as its standard process framework

The PMBOK Guide includes only the project management process area; it does not completely cover the other software engineering process areas that are all part of the software development process. We believe that the project management process area is the main process area within the software development process, and that the engineering process areas are support processes that the project management process area call at different moments.

We extended the PMBOK process framework in order to integrate process frameworks for each engineering process area in the software development process.

With this extended PMBOK process framework and the practices of a selected software process reference model, software engineering experts can detail all processes within the process framework, creating a meta software process definition.

RAMALA can have several meta software process definitions depending on the number of software process reference models stored in it.

Formal process definition

RAMALA uses the Entry Task Verification eXit (ETVX) process definition technique [10] which can be extended by adding more process definition elements that help us to obtain a meta process definition. The process definition elements that RAMALA uses are:

- 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
- Notes

In order to enhance process definitions, RAMALA permits process assets of any software process development methodology to link to some process elements (in red), i.e. RAMALA, as a knowledge base, gathers and classifies process assets such as templates, documents, or metrics of different software process development methodologies such as Prince 2, METRICA 3, RUP, etc.,

and links them to the corresponding process elements. RAMALA provides organizations with these process assets in order to adapt or improve their own process assets.

2.2 Definition and assessment of the software process in an organization

In order to enable organizations to assess and define their current software process, we felt that RAMALA had to fulfill the following requirements:

1. Formal software assessment method
2. Process asset manager

Formal software process assessment method

To determine the actual capacity of the software process of an organization according to a certain software process reference model, a formal assessment method for the software process reference model selected has to be used. Organizations can choose from several software process reference models stored in RAMALA's knowledge base to determine the capacity of their software processes. This means that there must be at least one formal assessment method for each software process reference model stored in RAMALA, making RAMALA a complex tool. A Formal Approximation for Software Process Improvement method [2] was implemented in RAMALA. This is a generic assessment method that covers any software process reference model stored in RAMALA.

The assessment result will be a color snapshot of the meta process definition of the selected software process reference model where colors reflect the degree of fulfillment of each process element.

Along with the assessment result, the organization will obtain the definition of their standard software processes that will be a subset of the RAMALA meta process definition of the selected software process reference model.

Process assets manager

During assessment, RAMALA gathers and classifies all direct evidences that indicate the implementation of the selected software process reference model practices, thereby creating the process assets repository of the organization, i.e. all documents and templates are gathered during assessment and linked to the corresponding process elements within the set of standard software processes of the organization.

Process assets of different software development methodologies that can be used to adapt or improve their own process assets will also be available to organizations.

2.3 Tracking of software process improvements

In order to ensure that new implemented software processes are institutionalized within the organization, we think that RAMALA has to satisfy the following requirements:

1. The project's defined processes mechanism
2. Process improvement tracking mechanism

Projects' defined processes mechanism

Once the organization's set of standard software processes is established, it has to be improved continuously according to the results of its own projects which determine the processes' strengths and weaknesses.

For each project, RAMALA allows the organization to establish the project's defined processes that will be a subset of the organization's set of standard software processes. Project results and documents will be stored in RAMALA as instances of the corresponding organization's process assets. In this case, RAMALA will also act as a historical database that helps project managers manage current and future projects.

By analyzing project results stored in RAMALA, software process improvement plans can be developed and later implemented.

Process improvement tracking mechanism

Once software improvement plans have been implemented, it is necessary to verify that improvements have really been implemented and followed. RAMALA helps organizations in this respect by querying and comparing instances of the project's process assets; gathering and analyzing measure data.

3. HOW TO USE RAMALA KNOWLEDGE BASE

RAMALA's most important features are described in this section.

RAMALA software uses the Application Service Provider (ASP) concept where software organizations only need an Internet browser and an Internet connection. Figure 2 shows RAMALA software architecture.

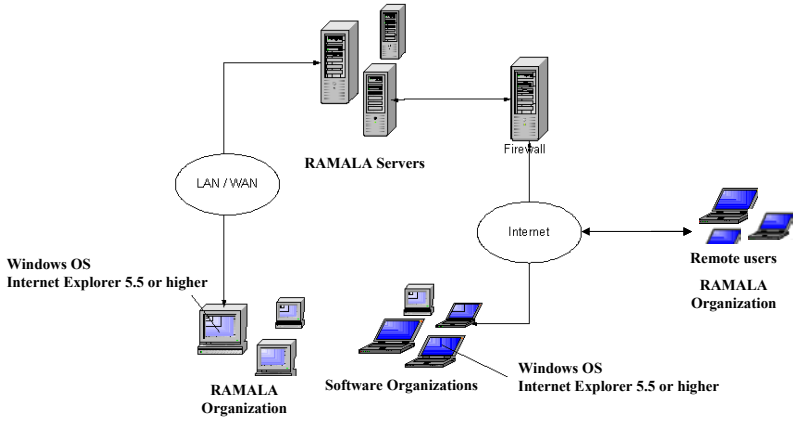

Figure 2: RAMALA software architecture

Software organizations can tour the knowledge base before selecting RAMALA. Once they sign on, a software organization has to first select a software process reference model. Currently, RAMALA has the CMMI and ISO 15504 models stored in its knowledge base. Figure 3 shows elements of the CMMI as an example of a software process reference model.

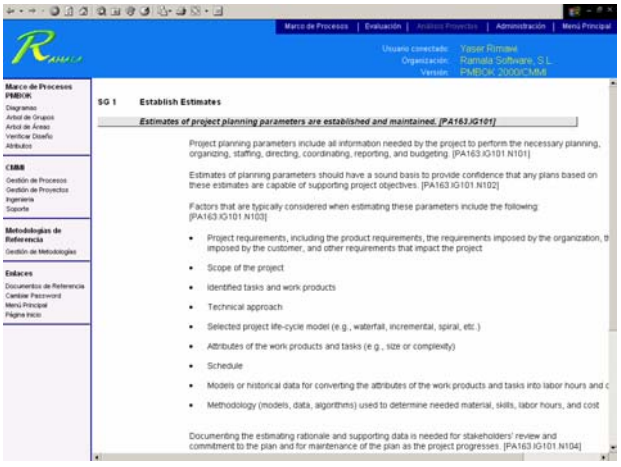

Figure 3: Software process reference model elements stored in RAMALA

As described earlier, RAMALA has stored a meta software process definition, based on the PMBOK process framework, for each software process reference model. The next step is to select

a set of processes to be assessed. Figure 4 shows how processes are selected for assessment in RAMALA.

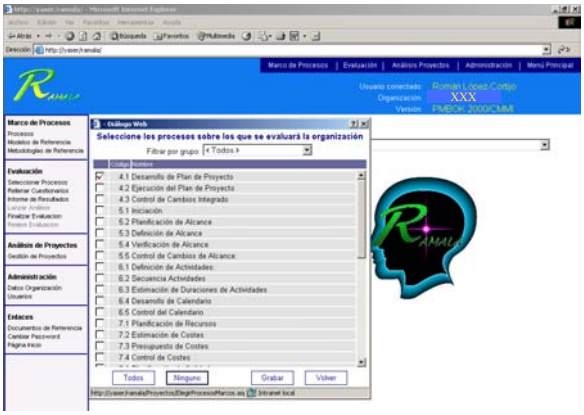

Figure 4:Selecting processes from the PMBOK software process framework for assessment

In order to assess the software process, special members of the organization have to complete a detailed questionnaire for each process and its elements selected. During the assessment, direct evidences (process assets of the organization) which indicate that the organization is satisfying the software process reference model practices, are collected, classified and linked to the corresponding software process elements and stored within the organization's own knowledge base. Figure 5 shows how the members of an organization have to complete the questionnaire for each process element, and how the organization's process assets are collected and linked to process elements.

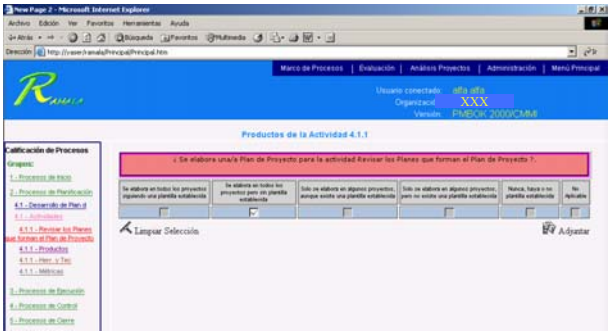

Figure 5: Process element assessment questionnaire

On completion of the questionnaire, an automatic algorithm, which calculates the capacity of each process and its elements, is executed. Figure 6 shows a report with the process elements capacity.

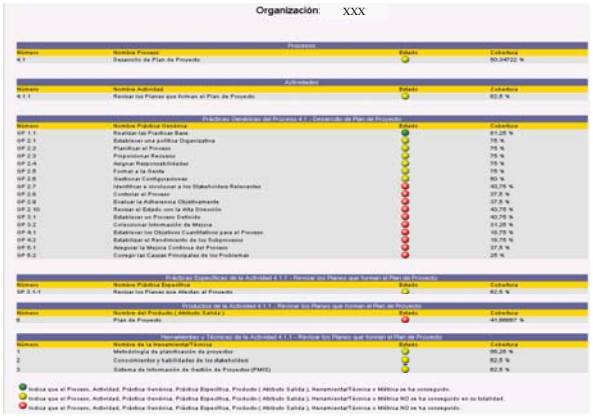

Figure 6: Software process elements capacity of an organization

Together with the assessment results, the organization will obtain its own knowledge base in which the definition of its set of standard software processes is stored as a color snapshot of the meta software process.

The organization can later manage its own knowledge base by adapting its process assets. RAMALA offers process assets of the most outstanding software process development methodologies that the organization can use to adapt their own process assets. Figure 7 shows an organization's process description stored in its knowledge base.

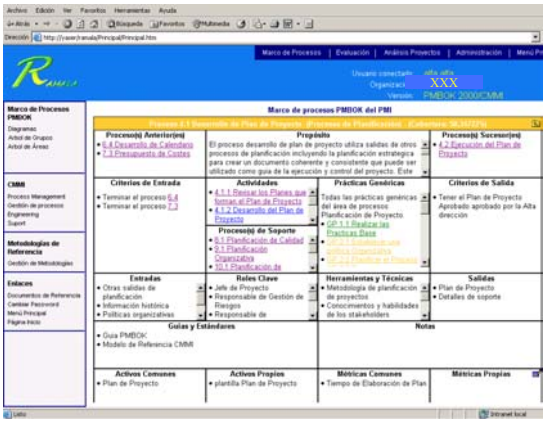

Figure 7: Standard software process of an organization

Once the organization has implemented a software process improvement plan based on the assessment results, RAMALA helps organizations assure the institutionalization of the new processes by acting as a historical database of software process assets instances of an organization's projects. An organization that uses RAMALA can:

- Create projects.
- Establish the project's defined processes for each project.
- Gather project results (process assets instances) and associate them with the corresponding project's defined process elements.
- Analyze project results.
- Determine the degree of fulfillment of new processes implemented.

Figure 8 shows how RAMALA establishes the project's defined processes and gathers the results, while figure 9 shows how RAMALA analyzes the results and determine the degree of fulfillment of new processes implemented.

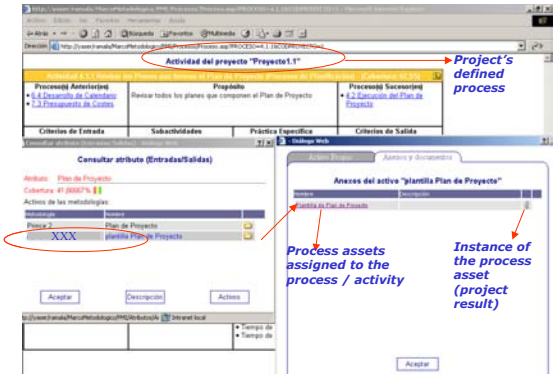

Figure 8: Historical database for project results of an organization

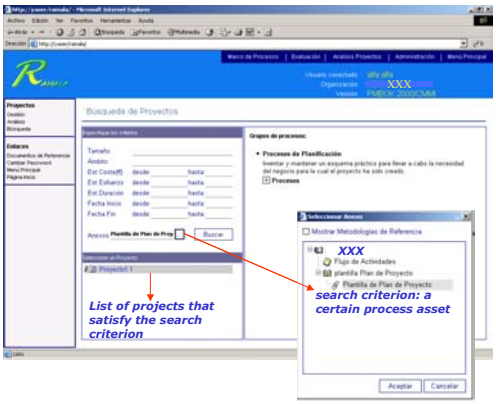

Figure 9: Assurance mechanisms for institutionalizing software processes

4. BENEFITS OF USING THE RAMALA KNOWLEDGE BASE

Software organizations, especially small ones, can obtain a lot of benefits from using the RAMALA knowledge base. Some of these are listed below.

- Reduced costs in the assessment and definition of the organization's set of standard software processes. RAMALA offers a simple formal software process assessment method through which an organization can obtain its software process definition that can be easily maintained and updated.
- The knowledge base of an organization (process assets) can be gathered, classified and associated with the corresponding process elements.
- The maturity of the software processes of an organization will be faster and easier because organizations will have:
    - software defined processes
    - available software process assets of the most outstanding software process development methodologies.
- A manageable historical database for software process assets instances providing good indicators on the degree of institutionalization of the organization's software processes.

CONCLUSION

In this work, we have presented the RAMALA knowledge base which contains all the necessary knowledge to carry out all the software process improvement activities. RAMALA permits:

- The assessment and definition of an organization's set of standard software processes with respect to the most outstanding software process reference models such as CMMI [11], ISO 15505 [3], and the most important project management standard: the PMBOK Guide [9].
- All the software process development knowledge of the organization (process assets) to be gathered and linked to with the corresponding process elements.
- A software process organization to have a software development thesaurus to reuse methodologies, standards, and products.
- The identification of the processes and activities needed to be carried out on each project.
- The storage of project results in a historical database for reuse on future projects.
- The institutionalization of software processes.

ACKNOWLEDGEMENT

Several prominent people in the business and academic world have made contributions to the RAMALA knowledge base. We would like to acknowledge their efforts and thank them for their contributions, especially the RAMALA team (Rimawi-Amescua-Mariscal-Andujar-Lopez-Andujar) who made RAMALA a reality.

REFERENCES

[1] Department of Defense of United States, "*Report of the Defense Science Board Task Force on Military Software*", Office of the Under Secretary of Defense for Acquisition, Washington, D.C. September1987.

[2] Javier Garcia, "*Formal Approximation for Software Process Improvement"* Ph.D. Thesis, Carlos III University of Madrid, November 2001.

[3] International Organisation for Standardization and International Electrotechnical Commission. "*ISO/IEC 15504 Software Process Improvement and Capability Determination Model (SPICE)"*, 1997.

[4] J. Herbsleb, A. Carleton, J. Rozum, J. Siegel, and D. Zubrow, "*Benefits of CMM-Based Software Process Improvement: Initial Results (CMU/SEI-94-TR-013)"*. Software Engineering Institute. Carnegie Mellon University, August 1994.

[5] M. C. Paulk, B. Curtis, M. B. Chrissis, and C. V. Weber, "*Capability Maturity Model for Software, Version 1.1 (CMU/SEI-93-TR-024)"*, Software Engineering Institute, Carnegie Mellon University, 1993.

[6] M. C. Paulk, B. Curtis, M. B. Chrissis, and C. V. Weber, "*Capability Maturity Model for Software, Version 1.1 (CMU/SEI-93-TR-025*), Software Engineering Institute, Carnegie Mellon University, 1993.

[7] P. Naur, and B. Randell, "*Software Engineering: Report of a conference sponsored by the NATO Science Committee*", NATO Scientific Affairs Division, Belgium, October 1968.

[8] P. Roshan, "*The Cost of CMM in a Conventional IT Organisation: A Field Study"*, Ph.D. Thesis, University of Detroit Mercy, 2002.

[9] Project Management Institute. "*A guide to the project management body of knowledge (PMBOK)"*, ISBN: 1-880410-22-2, 2000.

[10] R. Radice, N. Roth, Jr. O'Hara, and W. Ciarfella, "*A Programming Process Architecture"*. IBM Systems Journal, 24(2), pp 79-90, 1985.

[11] 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.

[12] Software Engineering Institute. "*Process Maturity Profile CMMI V1.1 SCAMPI V1.1 Appraisal Results 2004 Year End Update"*, Carnegie Mellon University, March 2005.

[13] Yaser Rimawi, *"RAMALA: A Model for Software Project Management Process Improvement"* Ph.D. Thesis, Carlos III University of Madrid, September 2004.