

RAMALA: A Knowledge Base for Software Process Improvement

Javier Garcia, Yaser Rimawi, Maria Isabel Sánchez, and Antonio Amescua

Computer Science Department, Carlos III University of Madrid, Avda. de la Universidad 30,
28911 Leganes, Madrid, Spain
jgarciag@inf.uc3m.es, yrimawi@gmail.com, misanche@inf.uc3m.es,
amescua@inf.uc3m.es

Abstract. The actual situation of small software organizations in software process definition and improvement is chaotic. Actually, deploying a software process improvement program within such organizations is very difficult, due to its high cost and small ROI percentage that could be obtained. RAMALA is a knowledge base, supported by a software tool called also RAMALA, that contains a software process framework, which is mainly based on the PMBOK process framework [9], detailed by software engineering experts using the best practices of the main software reference models like CMMI [11] and ISO 15504 [3], and enriched with process assets of the most outstanding software development methodologies. RAMALA is a platform where best practices of any software engineering process are recollected in a process definition form. Small software organizations can define, assess and improve their software processes economically using RAMALA.

1 Introduction

Software production in most small software organizations is characterized by poor management and individual skills, which are typical features of the “Software Crisis”. These features cause serious problems, such as, project delay, high costs, and poor quality products.

Software community is trying to deal with this problem since almost four decades. In 1968, Nato held a conference in which the term “Software Engineering” was born [7]. Since then, Software Engineering is a growing discipline. Two basic movements have enriched this discipline: technological and process. Programming languages, software tools and techniques characterize the technological movement while the process movement is focused on software process improvement elements and their support activities.

Within the last decade, the process movement has expanded widely in numerous software organizations where it has been proved that the major software problems are due to the 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 contribute to the process movement developing reference models and standards like CMM [5] [6], CMMI [11], ISO 15504 [3], and PMBOK [9]. Although there are several software reference models and standards that software organization can implement in order to improve their software processes, few organizations apply them. For example, in the world, just 567 organizations have conducted SCAMPI v1.1 class A appraisals from April 2002 through December 2004 [12]. This small number of organizations is due, among other reasons, to the high cost associated to deploy a software process improvement program.

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 engaged in software process improvement 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 of Improvement Project Effort
<u>CMM Process Flow Specification</u>	<u>19.90%</u>
<u>CMM Control Flow Specification</u>	<u>13.92%</u>
<u>CMM Data Flow Specification</u>	<u>11.53%</u>
Decision Maker Management	26.70%
<u>Product Related Process Assurance Activity</u>	<u>22.29%</u>
Initial Training	3.48%
On going training	2.18%

As we can see from the table, the first three activities (underlined) are related to process definition and their costs are more than 45% of the total, the cost of the process assurance activity (also underlined) is more than 22%. The cost of these four activities makes up 67.64% of the total cost.

This means that the major part of the cost of deploying the software reference model is the cost of software engineering experts. We believe that this percentage can be reduced, using a knowledge base supported by a software tool where the expert's knowledge can be gathered and managed.

RAMALA knowledge base, supported by a software tool called also RAMALA, gathers the software engineering knowledge needed to deploy a software process improvement program within a software organization. RAMALA knowledge base contains a software process framework, which is mainly based on the PMBOK [9] process framework, detailed by software engineering experts using the best practices of the main software reference models like CMMI [11] and ISO 15504 [3], and enriched with process assets of the most outstanding software development methodologies.

RAMALA knowledge base fulfills three main functionalities in a software process improvement deployment program: process assessment, process definition, and process improvement tracking.

2 RAMALA Knowledge Base

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

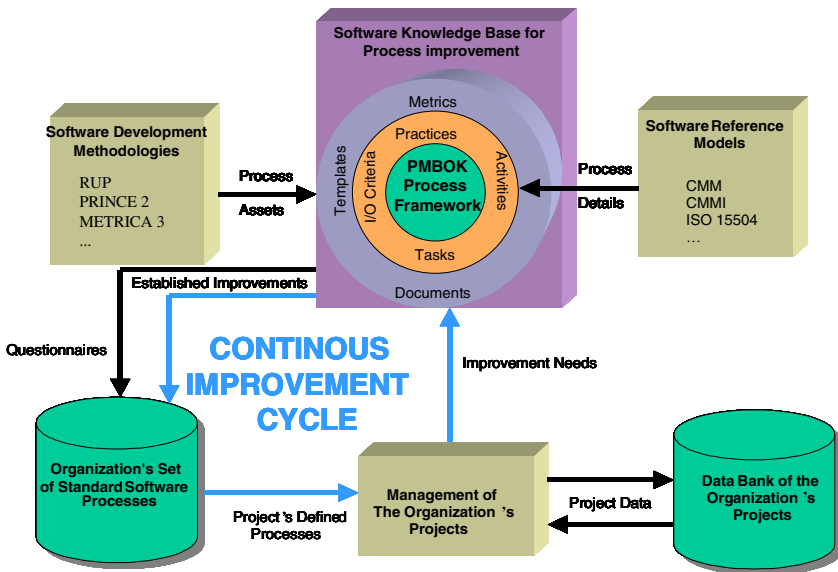


Fig. 1. RAMALA knowledge base structure

RAMALA knowledge base structure is shown in Figure1. As we can see, the process definition functionality is covered by the software knowledge base for process improvement component, where the PMBOK Guide Process Framework [9] is its core. Software engineering experts using the best practices of the software reference models and process assets of the most outstanding software development methodologies detail the process framework.

Implementing a formal assessment method valid for any software reference model covers the process assessment functionality. During the assessment, RAMALA gathers and classifies all 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 organization's set of standard software processes.

The improvement tracking functionality is covered by providing a mechanism to establish the project's defined processes, managing the project's process assets instances, and gathering measure data to verify the fulfillment of the improvements.

RAMALA knowledge base is described in more details in the following sections.

2.1 Software Engineering Knowledge Base for Software Process Improvement

In order to build a standard and robust software engineering knowledge base for software process improvement, we think that it had to satisfy the following requirements:

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

2.1.1 Standard Structure for Software Reference Models

RAMALA uses a generic data model where it is able to save in one repository all the elements of each software reference model; in this way organizations could have in one tool several reference models like CMM [5] [6], CMMI [11], and ISO 15504 [3].

2.1.2 Standard Process Framework

Software reference models help organizations to define their software processes, but 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 offers a process framework where all the necessary processes to manage any project are identified with all their dependencies. RAMALA uses the PMBOK Guide as its standard process framework

The PMBOK Guide just cover the project management process area, it does not cover in detail the rest of engineering process areas involved in the software development process. We believe that the main process area within the software development process is the project management process area, and that engineering process areas are support processes that the project management process area uses in different moments.

What we do is extend the PMBOK process framework integrating process frameworks for each engineering process area involved in the software development process.

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

RAMALA has several meta software process definitions according to the number of software reference models stored in it.

2.1.3 Formal Process Definition

RAMALA uses the Entry Task Verification eXit (ETVX) definition process technique [10], which has the advantage that it 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 enrich process definitions, RAMALA permits linking process assets of any software development methodology to some process elements (those underlined), i.e. RAMALA as a software engineering knowledge base gathers, and classifies process assets like templates, documents, or metrics of different software development methodologies such as Prince 2, METRICA 3, RUP, etc., and links them to the corresponding process elements. RAMALA provides these process assets to organizations to adapt them or improve their own process assets.

2.2 Definition and Assessment of Organization's Software Process

In order to enable organizations obtain an assessment and a definition of their actual software process, we think that RAMALA had to fulfill the following requirements:

1. Formal software assessment method
2. Process asset manager

2.2.1 Formal Software Assessment Method

In order to determine the actual capacity of the organization's software process according to a certain software reference model, we have to use a formal assessment method that covers the selected software reference model. RAMALA has stored in its knowledge base several software reference models that organizations can select to determine the capacity of their software processes. This means that there must be at least a formal assessment method for each software reference model stored in RAMALA, which makes RAMALA a complex tool. To solve this problem, RAMALA implements the Formal Approximation for Software Process Improvement method [2]. This is a generic assessment method that covers any software reference model stored in RAMALA.

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

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

2.2.2 Process Asset Manager

RAMALA, during the assessment, gathers and classifies all direct evidences that indicate the implementation of the selected software reference model practices creating the organization's process assets repository, i.e. all documents and templates are gathered during the assessment and associated with the corresponding process elements within the organization's set of standard software processes. Also, organizations will have available process assets of different software development methodologies that can use to adapt or improve their own process assets.

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

2.3 Tracking of Implemented Software Process Improvements

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

1. Project's defined processes mechanism
2. Process improvement tracking mechanism

2.3.1 Project's 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, where project results determine the processes' strengths and weaknesses.

For each project, RAMALA allows the organization 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 an historical database that helps project managers manage current and future projects.

Analyzing projects results stored in RAMALA, software process improvement plans can be developed and later implemented.

2.3.2 Process Improvement Tracking Mechanism

Once software improvement plans are implemented, it is necessary to have evidence that improvements have really been implemented and followed. RAMALA helps organizations in this aspect by making queries and comparisons on instances of the project's process assets, gathering and analyzing measure data in order to track the improvement fulfillment.

3 How to Use RAMALA Knowledge Base

The most important features in using RAMALA will be described in this section. RAMALA software applies the Application Service Provider (ASP) concept, where what software organizations only need is an Internet browser and an Internet connection. Software organizations, before signing on RAMALA, can make a tour

within the knowledge base. Once, they sign on, the first thing that a software organization had to do is selecting a software reference model that wants to follow. Actually, RAMALA knowledge base has stored CMMI and ISO 15504 models. Figure 2 shows elements of the CMMI as a selected software reference model.

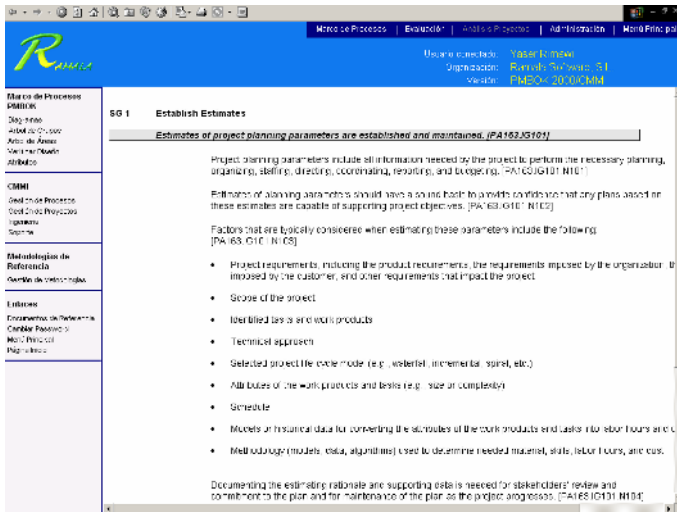


Fig. 2. Software reference model elements stored in RAMALA

RAMALA, as described before, has stored for each software reference model a meta software process definition based on the PMBOK process framework. The relevant next step that the software organization should do is selecting a set of processes that wish to assess. Figure 3 shows how processes are selected for assessment in RAMALA.

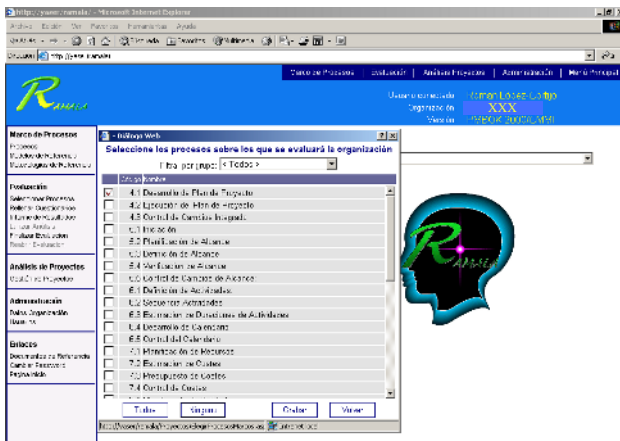


Fig. 3. Selecting processes from the PMBOK process framework for assessment

In order to carry out the assessment, special members of the organization had to fulfill a detailed questionnaire for each process selected and its elements. During the assessment, direct evidences (organization’s process assets), which indicate that the organization is satisfying the software reference model practices are collected, classified, associated to the corresponding software process elements, and stored within the organization’s particular knowledge base in RAMALA. Once the organization finish filling questionnaires, an automatic algorithm is executed, which calculates the capacity of each process and its elements. Figure 4 shows a report with the process capacity.

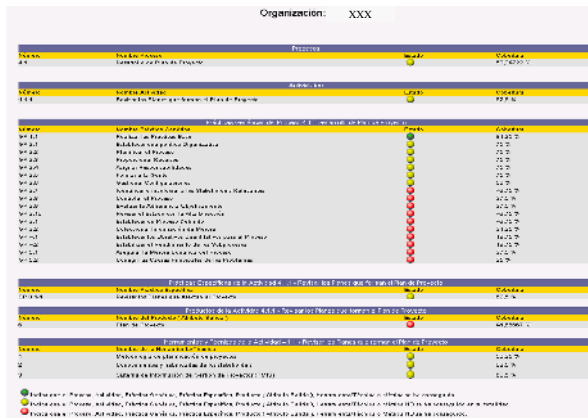


Fig. 4. Organization software process capacity

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

Later, the organization can manage its own knowledge base adapting its process assets. RAMALA offers process assets of the most outstanding software development methodologies that the organization can use to adapt their own process assets. Figure 5 shows an organization’s process description stored within its knowledge base.

Once the organization implements a software process improvement plan based on the assessment results, RAMALA helps organizations assure the institutionalizing of the new processes acting as an historical database of organization projects’ software process assets instances. An organization that uses RAMALA can:

1. Create projects
2. Establish the project’s defined processes for each project
3. Gather project’s results (process assets instances) and associate them to the corresponding project’s defined process elements.
4. Analyzing project’s results
5. Determine the fulfillment degree of new implemented processes.

Generally speaking, it is important to indicate that everyone surveyed knew the improvement models and they all appreciated that the approach and focus of the RAMALA model offered significant improvement over the rest of the models they know.

In every case, they also considered that the process defined in RAMALA for process definition is useful and easy for top management and strategic consultants who are responsible for defining the organization's strategies.

With regards to the use of the knowledge base as a support for the RAMALA model, top management and the strategic consultants evaluated positively (85%) the possibility of defining quantifiable objectives for process improvement and providing metrical objectives for the organization's software process improvement in real time.

To confirm the validity of the results, the disparity between the answers was studied. As the disparity in the answers obtained is 0.33 out of 1.2 (less than 30%), so we can state that the level of dispersion is low, and, consequently, we can consider the results to be representative.

- At the tactical level because middle management plan, control and track the organization's different software projects undertaken. At this level RAMALA provides efficient mechanisms for software project management process definition and assessment of the efficiency and quality of the work procedures to carry out the projects mentioned.

Therefore, at this level, the aim of the model was to gather evidence to ensure:

- the process representation capacity with RAMALA.
- the evaluation capacity of the organization's current practice with RAMALA.

The team leaders responsible for improvement or the experts in software improvement processes who had used the RAMALA tool filled in questionnaires in order to gather evidence. Eleven questionnaires were registered.

To assess the results of the capacity verification of the RAMALA tool, the data gathered from the evaluation questionnaires filled in by the 11 improvement team managers or the software improvement experts were analyzed.

All of them were familiar with software improvement methods and all rated the RAMALA model positively. The representation capacity and software project management process assessment which the RAMALA model offers were analyzed separately.

At a general level, the representation capacity of the RAMALA software was considered to be high and capable of meeting the organization's objectives. In this respect, therefore, all the replies were rated HIGH or VERY HIGH. Only 10% of the replies were partially unfavourable.

The analysis of the RAMALA's capacity to assess the quality and efficiency of an organization's software project management processes were rated very high. At the same time, 37.5% of those surveyed thought that the results with RAMALA reflected reality and were very easy to understand, while 62.5% considered them useful, in general terms.

- At the operational level, in the field of research, the activities consisted of designing and implementing project management work procedures and instructions. Thus, at this

level, the aim of the model was to gather data to verify effort reduction RAMALA provides in the definition and implementation of project management processes.

During this validation phase, the researchers controlled the effort spent by seven groups in charge of defining and implementing a software project management process using RAMALA was analyzed. This data was obtained from seven teams, totaling 32 people. This information, obtained from set of workgroups, was checked against the information gathered from other workgroups that, in the previous months, had to define and implement a software project management process without using RAMALA. This control data was obtained from six teams, grouping 33 people.

The effort data used for this validation was accounted using an effort registry form (specially designed for this purpose) that had to be filled in each week by each member of the working groups. Table 2 shows the effort the software project management processes definition and implementation work teams accumulated.

Table 2. Effort of Software Process Improvement activities

	Number of Groups	Time (minutes)	Average (minutes)
Without Ramala	6	62099	10349,8
With Ramala	7	44706	6386,5

Analyzing table 2, we can say that RAMALA reduces costs in assessing and defining the organization’s set of standard software processes. RAMALA offers a simple formal software process assessment method, where as a result, the organization can obtain its software process definition that can be maintained and updated easily.

5 Conclusion

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

- Assess and define the organization's set of standard software processes with respect to the most outstanding software reference models like CMMI [11], ISO 15505 [3], and the most important project management standard: the PMBOK Guide [9].
- Gather all the software development knowledge of the organization (process assets) and associate them with the corresponding process elements.
- Provide software organizations with a software development thesaurus to re-use methodologies, standards, and products.
- Identify for each project the processes and activities needed to be carried out.
- Store all the project’s results in an historical database in order to be used in future projects.
- Assure software process institutionalizing.

References

- [1] Dept. of Defense United States, “*Report of the Defense Science Board Task Force on Military Software*”, Office Secretary of Defense for Acquisition, Sept.1987.
- [2] Javier Garcia, “*Formal Approximation for Software Process Improvement*” Ph.D. Thesis, Carlos III University of Madrid, November 2001.
- [3] International Organisation for Standardization. “*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. 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, 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, 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*”, 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.