

# Using a qualitative research method for building a software maintenance methodology



Macario Polo<sup>\*,†</sup>, Mario Piattini and Francisco Ruiz

*Grupo Alarcos – Department of Computer Science, University of Castilla-La Mancha, Paseo de la Universidad, 4, 13071-Ciudad Real, Spain*

---

## SUMMARY

This article explains our experience of using Action Research to develop a software maintenance methodology involving two organizations: a group of university researchers and a software services organization. The concept of ‘methodology’ comprises a wide set of elements whose identification, definition and integration is not a trivial task, due to the magnitude of the project and to the different nature of the organizations. The use of Action Research was a key factor in the progress of the research and has been essential in the adoption of the methodology within the software services organization. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: Action Research; process models; qualitative research; software maintenance

## 1. INTRODUCTION

Software maintenance is the main source of work for software organizations [1]. In spite of this, the majority of them have their structures, methodologies and work teams prepared for new developments: Singer [2] reveals that 61% of the professional life of programmers is dedicated to maintenance and only 39% to new developments. Van Bon [3] notes that the management of the software maintenance process is and has been neglected for many years, which has led to a lack of specific management methodologies. Moreover, methodologies for software development are not directly applicable to maintenance, since they disregard future stages of maintenance [4]. Basili *et al.* [5] confirm this issue, affirming that the definition and validation of methodologies that take the specific characteristics of software maintenance organizations and of their processes into account are required.

---

\*Correspondence to: Macario Polo, Escuela Superior de Informática, Paseo de la Universidad, 4, 13071-Ciudad Real, Spain.

†E-mail: Macario.Polo@uclm.es

Contract/grant sponsor: MANTIS; contract/grant number: CICYT 1997-1608 TIC

Contract/grant sponsor: MPM; contract/grant number: PROFIT 070000-2000-307

---

Therefore, there is a general agreement that the characteristics of software maintenance are so different from those of development that many of the development techniques, tools, model processes, etc. are not directly applicable to maintenance.

- One possible cause of this is the difference in effort which must be devoted to tasks of development and maintenance: it is common practice among programmers to put the majority of their effort into the coding and unitary testing tasks during development; in maintenance, most effort is devoted to understanding code and to non-regression testing.
- Another possible reason for such a difference has been suggested by [6], for whom maintenance has more similarities with service than with software development, thus implying the avoidance of the application of common development strategies to maintenance.

It may be that such divergence has been what has motivated an increasing number of software companies to offer software maintenance services. Taking into account that, at present, a general lack of control in the maintenance process exists, a well-managed and controlled maintenance process could be offered by third-party organizations as an attractive value-added service. This activity is one of the main sources of business of our partner company, Atos ODS, the third European organization, as regards business volume, in software and telecommunication services. This company carries out the software maintenance of large public, banking, industrial and telecommunications corporations.

The aim of this paper is to present and share our experience in the application of a qualitative research method for building Mantema, a methodology for software maintenance which has been jointly developed by our research group at the university and by Atos ODS. We are not, therefore, presenting the methodology itself, a description of which may be found in [7]. We believe that sharing our experience could be positive for both researchers and practitioners, due to the characteristics of the project and to the different nature of the two organizations involved. This caused us to search for the most adequate method to define and put the methodology into practice, and we considered Action Research to be a very acceptable candidate. We knew the suitability of Action Research for performing qualitative research with the involvement of several parties (in other areas, such as organizational development) thanks to the papers referenced by [8]. Later, [9] confirmed the applicability of Action Research for information systems. More recently, qualitative methods in general (including Action Research) have been accepted for carrying out research in software engineering [10].

Currently, most of the empirical research undertaken in software engineering is quantitative and statistics based; however, we believe that qualitative and observational studies should be encouraged as they can also be valuable sources of knowledge.

The paper is organized as follows: Section 2 provides a brief description of Action Research; in Section 3, we explain the concept of 'methodology', describing some of its implications; Section 4 describes how Action Research was used to build the Mantema methodology. In Section 5 we present our conclusions and future research work. Appendix A contains the main milestones of the project and Appendix B a description of the pilot projects used to build and test the methodology.

## 2. A SHORT DESCRIPTION OF ACTION RESEARCH

Action Research is a qualitative research method that brings theory and practice, and researchers and practitioners together to resolve a problem [9]. According to these authors, Action Research is particularly suitable for research that includes real organizations.

Qualitative research differs from quantitative research in a number of aspects. These vary from its purpose (to provide insights into the setting up of an organization or group of people, the problem and its solutions versus the generalization of results from a sample of the population of interest), the sample size (usually a small number versus a large one), the data collection method (unstructured or semi-structured interviews, observations and group discussions versus structured techniques) and the data analysis mechanism (non-statistical versus statistical).

In the field of information systems, the research client is an organization for which the researcher provides various services, such as consultancy or change management, in return for access to research data [11].

More formally, the actors involved in the research process of Action Research are [12]:

1. The *Researcher*, the individual or group who actively carries out the research. In our case, the Alarcos Research Group.
2. The *Critical Reference Group*, which is the person/group with the problem for which the Researcher must find a solution. The Critical Reference Group also participates in the research process, although less actively than the Researcher. In a strict sense, it was the Atos ODS company which had the problem which needed to be resolved, so could be assumed to play this role. More concretely, only the people from Atos ODS that had interacted with the Researcher really constituted the Critical Reference Group; that is to say that, during the first months of research, only the managers of the Maintenance and Outsourcing Unit took part. When the project was more advanced, more people (programmers and project managers) directly involved in the pilot projects took part in it.
3. The *Researched Object* is the problem to be solved and, in this case, it is the management of the software maintenance process. A possible way of achieving better management (that is, a solution to such a problem) is through the use of specific methodology.
4. The *Researched For* is usually an external actor, not directly involved in the research, but who might benefit from better information regarding the problem. In the Mantema project, Atos ODS's customers have this role, and there may also be other software organizations which could obtain benefits from the methodology.

Action Research can be carried out in various ways. French and Bell [8] distinguish four variants which depend mainly on the characteristics of the research project.

1. *Diagnostic*: the Researcher studies a problematic situation, makes a diagnosis and some recommendations to the Critical Reference Group, without following up the results.
2. *Participatory*: the Critical Reference Group puts the recommendations made by the Researcher into practice, sharing their effects and results together.
3. *Empirical*: the Critical Reference Group has a broad and systematic log of its action and effects. This characteristic makes this variant very difficult to apply.
4. *Experimental*: several routes to reach the goal are evaluated. However, it is practically impossible to evaluate the different routes because if evaluations are made in different Critical Reference Groups or if they are made at different moments in the same Group, the distinct characteristics of the groups or of the moments mean that adequate conclusions cannot be drawn.

Participatory Action Research was selected as the most suitable variant for our project. Research carried out using this variant is eminently active, in the sense that there is continuous feedback between

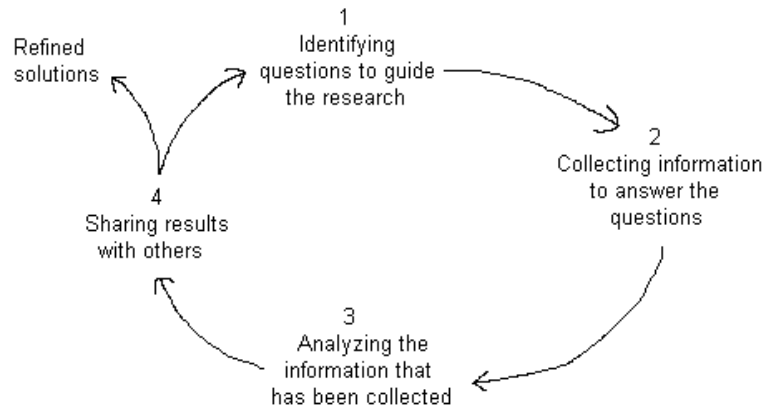


Figure 1. The cyclic process of Action Research.

the Researcher and the Critical Reference Group: the first looks for solutions for the Researched Object, whereas the second applies these solutions to its environment and explains the results to the Researcher [13]. As Figure 1 shows, a more refined solution is produced after each research cycle.

### 3. IMPLICATIONS OF BUILDING A METHODOLOGY

According to [14], the concept of ‘methodology’ in the context of software engineering is not trivial. In fact, among the elements that a methodology should provide, these authors cite the following:

1. a process model, which indicates the tasks we need to perform to execute the process;
2. metrics, to assess different quality attributes of the process and of the products generated;
3. a set of techniques, to help in the execution of some tasks of the process;
4. tools, which may automate tasks of the process and possibly its management;
5. deliverables, which should contain the definition of all the possible document templates to be used during the process;
6. guidelines for helping in the process management, such as role identification, definition of team structures, etc.

The building of a methodology that fulfills this wide concept implies the analysis, definition and integration of elements of these types, all of which are integrated into the process model, which is the core of the methodology. In fact, the process model tells us in *which* tasks we need to use *which* metrics to measure *a certain* quality attribute; the process model also tells us *which* technique must be employed to carry out a task more easily. There are vertical tools (that probably automate manual techniques) to execute some tasks *defined* in the process model and horizontal tools which fulfill the *process model* for managing projects.

Therefore, building a whole methodology requires an expensive extended trial and error period. The fact is that many of the methodologies for software development have been built after many years of working and experience, sometimes as the union of other well-known, well-tested methodologies or parts of methodologies (for example, the definition of UML, considered as a part of the Unified Development Process, is defined by taking different pieces of the notation of OMT, Booch, etc., and people around the world are continuously proposing new stereotypes, constraints, extensions, etc. to the language). Experiences of other researchers in both our and other areas, and the need to avoid an overly long development period encouraged us to select Action Research to conduct this project.

#### 4. APPLYING ACTION RESEARCH

Atos ODS was responsible for the maintenance of large systems developed in Cobol, mainly in banking, industrial and telecommunications environments. In order to tackle the problem of maintaining these systems facing the Y2K and the Euro effect, the company saw the need to establish a new methodology that would allow the systematization of the maintenance tasks, and the substitution, interchange and incorporation of personnel without influencing the cost or time of the projects, and which would keep control of the process. With this goal in mind, Atos ODS contacted the Alarcos Research Group of the University of Castilla-La Mancha, which specialize in the definition and implantation of methodologies for the development of information systems and in the use of international standards in this area. Atos ODS and the University applied for a research grant to the, at that time, Minister of Industry and Energy, to develop a methodology for software maintenance. This project was approved by the Atyca program (grant number TA15/1999) and was carried out between July 1997 and December 1999.

During the first phase of the project, which could correspond to the first step of Action Research shown in Figure 1, we had several meetings at the offices of Atos ODS with the managers of its Maintenance and Outsourcing Unit (who, in this way, became the first members of the Critical Reference Group). These meetings served to describe, make contact and familiarize ourselves with the problem that Atos ODS had. They mainly explained:

- The context in which the Atos ODS Maintenance and Outsourcing Unit worked. Most of its customers are important companies with a large number of big legacy applications, initially written more than ten years ago mainly in Cobol, Cobol/CICS and DB2. The lack of qualified people to work in information systems companies, with a demand for qualified staff which is greater than the supply [15], meant that people with other types of qualification (physics or mathematics, and even others, such as Philosophy) were employed by the company as developers and maintainers. These people were trained for some weeks or months in the specific language and environment in which they were to work; after the course, they were usually moved into the customer offices to start their job. However, mobility of staff in all companies is high, constituting a serious problem, which is perhaps more acute in maintenance projects. On the other hand, the number of contracts and projects within the company was continually increasing, both at national and international level, which forced the company to incorporate new personnel.
- The way in which they are accustomed to carrying out their current maintenance projects. Most Atos ODS customers, as in most organizations with no maintenance methodology, solve

their problem by 'brute force', assigning as many people as they think are required to the problem. Unfortunately, these people work with undocumented systems, and in so doing become indispensable and not easily exchangeable, thereby causing serious difficulties and delays when they leave the project or the company. As a result, there is no systematic, organized process; they are continuously 'extinguishing fires', in a state of emergency, and are incapable of planning. Under these conditions, these organizations either maintain underused resources or burn them out, making them work above their possibilities. On the other hand, these organizations do not use metrics to measure the complexity of programs to be maintained, which leads to accurate estimations not being obtained. This has negative consequences, implying a lack of maintenance control.

With this initial information, collected via conversations whose semi-formal agenda was previously distributed to all the members of the Critical Reference Group and of the Researcher, we were able to obtain a good understanding of the Atos ODS situation. With these meetings, both parties agreed on the need to start looking for possible already existing solutions regarding the Researched Object (that is to say, a methodology for managing the software maintenance process) that could be directly applied by Atos ODS or, otherwise, tailorable at a reasonable cost.

#### 4.1. Looking for a process model

Most methodologies for software engineering areas share the afore-mentioned idea of having a core process model. For example, in the first lessons of one of the most recently referenced methodologies, the Unified Development Process, a spiral model is generally used to explain its main structure, stages and so on.

With this idea in mind, and placed in the second step of Action Research (Figure 1), the researcher started looking for process models for the maintenance process, which caused several related proposals to be studied, including standards and experience reports. The most meaningful of these could be used as reference frameworks to progressively complete the methodology with the rest of the elements presented in Section 3, such as the IEEE 1219 Standard for Software Maintenance [16], the ISO/IEC 12207 International Standard for Software Life Cycle Processes [17], the draft of the future ISO/IEC 14764 (which analyses the 12207 Maintenance Process in more detail) and some maintenance guides, such as those of [18].

The people that, at that moment, composed the Critical Reference Group, studied the selected and commented proposals prepared by the Researcher, which was an instance of the third step shown in Figure 1, for some weeks; they then drew up a descriptive document containing the characteristics of each of them. The initial preference of the Critical Reference Group was for the IEEE 1219 Standard, mainly motivated by the experience of one of its members, since he had experience in the application of IEEE standards during his years of service in the Spanish Army. However, in a decisive meeting, the Researcher was able to convince the Critical Reference Group of the greater suitability of ISO/IEC 12207 highlighting: (1) the big international impact that this international standard was having in the life cycle processes in general (Radatz *et al.* [19] report that several other standards are being substituted or modified due to the advent of the 12207), as in the particular case of maintenance ([1] notes that 'ISO/IEC 12207 will drive the world software trade and will impact on maintenance'); and (2) the Spanish Minister for Public Entities had adopted the 12207 Standard as the basis for the Metrica

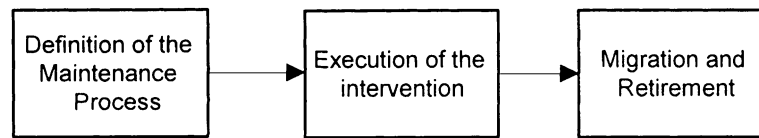


Figure 2. The macrostructure of the ISO/IEC 12207 maintenance process.

Methodology, which must be used to develop software for the Government [20]. Finally, both parties selected this standard, also influenced by the existence of a tailoring process in the same standard: the tailoring process makes possible to obtain an adapted version either of the standard or of a part of it more easily, remaining this version conforming to the standard itself.

#### 4.2. Defining the process model of Mantema

At this point, we had completed the first turn of the Action Research cycle and were, therefore, ready to 'identify questions to guide the research', although now with the way much clearer and with a much stronger basis than in the beginning. Note, however, that although this first turn produced a more refined solution of the Researched Object (as is expected from the reading of Figure 1), it was still not a directly applicable solution to the problem, but just a very concrete set of materials, a proposal and a common agreement to continue the work. In spite of this, this first turn was an essential factor for the success of the project.

From a more-in-depth study of the maintenance process defined in the 12207 Standard, we found that its six activities (process implementation, problem and modification analysis, modification implementation, maintenance review/acceptance, migration and retirement) could be visually grouped into a set of activities that compose a graph like that shown in Figure 2. Each one of its nodes constitutes a 'pluggable' piece of the whole process that can be independently analyzed and studied in such a manner that, at any given moment, we can focus our effort on just one fragment.

The most important node in Figure 2 is perhaps the central one, which contains the activities and tasks to be executed for the software modification, and which is mainly composed of the 'modification implementation' activity of ISO/IEC 12207. This activity introduces, as its tenth task, a complete execution of the 12207's development process, which in turn consists of 13 tasks. Table I depicts the results of this introduction.

To our understanding, maintenance with a development process inside involves an excessive number of tasks, some of which overlap (as we analyzed in [21]). At a new meeting whose purpose was to study the second node of Figure 2, the Researcher explained the possibility of deleting those tasks of the development process that are introduced in maintenance but which overlap or are not useful. There was an agreement on those that could be removed.

At this point, we took the decision to follow the idea of the 12207 Standard, i.e. dividing the future process model into activities, which are in turn composed of tasks. We then began to define a common structure for tasks, understanding at that moment that every task could be characterized by the corresponding activity in which it was included, a name, the input and output products, techniques

Table I. The modification implementation activity in ISO/IEC 12207.

Activity	Task No.	Task description	
...	1–8	...	
Modification implementation	9	Conducting analysis and determining which documentation, software units and versions need to be modified	
	10	Entering at the development process to implement the modifications	
		10.1	Process implementation
		10.2	System requirement analysis
		10.3	System architectural design
		10.4	Software requirement analysis
		10.5	Software architectural design
		10.6	Software detailed design
		10.7	Software coding and testing
		10.8	Software integration
		10.9	Software qualification testing
	10.10	System integration	
	10.11	System qualification testing	
10.12	Software installation		
10.13	Software acceptance support		
...	11–24	...	

that could help in its execution and the people in charge; that is to say, by some of the elements which, if adequately integrated, could continue composing the methodology, in the sense exposed in Section 3.

For this last issue (people responsible for each task), we merely made a distinction between the three companies involved in maintenance outsourcing projects. In the ISO/IEC 12207 IS there is also a set of definitions for possible organizations involved in software projects. In our case, we initially distinguished the following:

1. the Maintenance Organization, as the company that provides the software maintenance service;
2. the Customer, as the organization that owns the software under contract;
3. the User, as the organization that uses the software under contract.

As can be seen, it is not casual that the first actor identified plays the role of Critical Reference Group in our Research, whereas the Customer and the User are probably members of the possibly wide community of ‘Researched For’ actors.

#### 4.3. Refinement of the process model

Once the tasks of the ‘Execution of the intervention’ node of Figure 2 had been identified and selected, the Researcher gave all their attention to building an initial structure of a small process model for it, filling-in some of the elements that composed their tasks: for the moment only input and output products (some of them in the form of documents), and techniques. Four pilot projects were then selected by Atos ODS to apply to future, successive versions of these pieces of methodology, and of the methodology as a whole, which are described in Appendix B.



After some discussions with the managers of the Maintenance and Outsourcing Unit, mainly related to the contents of each task, the maintenance personnel in charge of the pilot projects received the first versions of the process model available at that moment, what included some programmers as project managers. These people were, therefore, becoming part of the Critical Reference Group. In fact, according to the definition of [12] reproduced in Section 2, they belong to the ‘group with the problem’, but until this moment they had not participated in the research process. Now they were starting to participate, although for the moment they did not directly share their impressions with the Researcher. Moreover, they still did not apply these first versions of the process model: they continued working as before (with their current method), but taking notes and making observations about its supposed application. Every week, and together with their habitual working reports, they reported their comments on the simulated application of the methodology to their respective project managers, who in turn explained their suggestions, observations, etc. to the members of the Critical Reference Group that acted as our interlocutors.

We met these interlocutors periodically and also, from the third meeting on, the person responsible for the quality of Atos ODS, since the company was to begin a certification process in ISO 9000. In this way, new methodology drafts were developed that were again delivered to the maintenance teams. At that moment in time, the results of the simulated application were given to us by our usual contacts rather than being directly reported to us by the final users, although we had copies of all their notes and observations, as well as all kinds of reports, summaries, etc., made by the project managers.

Therefore, in these first stages, the application of the methodology was simulated and not real, and consequently the opinions of the maintenance personnel did not totally correspond with those of a real application, and there could have been some misinterpretation due to the intermediate steps that the conversations took. This risk had to be assumed by all the parties, since the main goal of Atos ODS at that moment was to guarantee the operation of the systems being maintained. The personnel had such a heavy work load at the time that they could not carry out a parallel execution.

The process model became more mature when, as a result of the analysis of the results of these initial applications (in many cases, maintainers did not execute all the proposed tasks or, if they did, they added notes explaining the possible lack of suitability of some of them), the practical impossibility of considering a single set of tasks to be followed up for all the possible types of maintenance was evidenced, i.e. the nature of projects (and, within the projects, of interventions) were so different that it was not reasonable to have the same set of tasks for all the maintenance types. This occurred, for example, with the tasks proposed for analyzing alternative solutions and for carrying out integration tests, which were quite systematically disregarded in urgent modifications. Although, due to the privacy clauses signed by Atos ODS and its customers, we sometimes had difficulties in accessing concrete quantitative data on the projects, we observed later that most urgent interventions affected only a few lines of code that did not require this kind of test. On the other hand, it is quite obvious that this type of change requires neither a documented proposal nor analysis of alternatives.

With this situation, we held a meeting to propose the definition of different process models for the different types of maintenance in which the Critical Reference Group was interested. Initially, they identified those that they had experienced, but without defining them clearly; in fact, these types had only a broad, coarse correspondence with those usually identified in the literature. However, we built, for example, a first version of a process model for a type of maintenance that our interlocutors called ‘evolutive’. In spite of the initial conformity of Atos ODS with this definition, in reality the maintenance personnel never carried out all the tasks of this kind of intervention, since it really corresponded to a

mixture of adaptive and perfective, but without entirely corresponding to either of them. Figure 3 shows a table, corrected after a joint meeting, corresponding to the 'evolutive' type of maintenance that was initially identified but finally removed from the methodology, and that was divided into two different types: perfective and adaptive.

As a consequence of this kind of feedback, the Researcher proposed distinguishing different process models for corrective, perfective, preventive and adaptive maintenance types (which are the types usually distinguished in the literature), together with a more rigorous characterization of them. The execution of different tasks depending on the maintenance type of the modification request is a recommended practice by some references [16,17,22], and the idea was quickly accepted by our interlocutors at the Critical Reference Group. Such a distinction was a very important step in the project, since we could focus our main effort on the development of a brief process model for each type of maintenance, thus producing a set of 'technical guides', one for each type, which had also been suggested by the members of the maintenance teams. Perhaps, for the reason of simplicity, corrective maintenance received our main initial attention.

The first technical guide released (that of corrective maintenance) started its real application in the pilot projects about seven months after the project was signed (see the table with the main milestones in Appendix A). For the data collection and monitoring of the maintenance projects using the first versions of Mantema, Atos ODS adapted the MANUTEN CONDUCT tool that had previously been used by Atos ODS both in international (with Fiat Auto) and national projects (Valencia City Council or the Argentaria Banking Corporation), as shown in [23]. Figure 4 shows one of the reports about corrective maintenance obtained by MANUTEN CONDUCT. As can be seen in the figure, corrective maintenance interventions were at that moment controlled differently according to three different categories: very critical, critical and non-critical. Later, it was demonstrated that the 'very critical' and 'critical' anomalies could be jointly managed by a new type of maintenance, the 'urgent-corrective', because all the interventions of these types followed the same set of tasks. This fact evidenced, as one of the first results of the actual application of the methodology, the convenience of dividing the corrective maintenance into two subtypes depending on the urgency of the intervention, and thereby obtaining the urgent and the non-urgent corrective maintenance types.

We were, therefore, dealing with five types of maintenance (urgent corrective, non-urgent corrective, perfective, preventive and adaptive), each one with its own set of activities and tasks. However, mainly motivated by the different economic treatment that Atos ODS gives to each type of intervention, the last four types were grouped under the 'plannable maintenance' denomination, the urgent-corrective remaining as 'non-plannable'. This distinction is just conceptual, since there really exist five types of maintenance with their five corresponding process models. After this change, the initial structure of Figure 2 was redrawn as in Figure 5, which was used during the joint meetings to abstract ourselves from details.

The distinction of five actual maintenance types (although conceptually grouped into two types) facilitated more members of the Critical Reference Group to play a more active role in the research process, since the maintenance teams could really apply the different technical guides to the pilot projects and report their actual results, without breaking contact with our initial interlocutors. So, for example, the analysis of the conversations and results reported by programmers evidenced that the non-urgent corrective and the perfective maintenance types shared exactly the same set of activities and tasks, which made us give a common denomination to both types. This proposal was initially accepted by the managers of the Maintenance and Outsourcing Unit, but the later incorporation of outsourcing

**ACTIVIDADES Y TAREAS EVOLUTIVO**

*Correctivo no bloqueante*


Entradas	Análisis del error:					Intervención correctiva					Cierre intervención	
	1.1	1.2	1.3	1.4	1.5	2.1	2.2	2.3	2.4	3.1-3.5		
Producto software en explotación con error no bloqueante y petición de modificación	Investigar y analizar causas	Documento diagnóstico y posibles soluciones	Elementos software con errores visibles	Elementos software con errores visibles	Elementos software con errores visibles	Elementos software con errores visibles	Elementos software con errores visibles	Elementos software con errores visibles	Elementos software con errores visibles	Elementos software con errores visibles	Elementos software con errores visibles	Elementos software con errores visibles
Salidas	Elementos software a corregir	Causa del error	Causa del error	Causa del error	Causa del error	Causa del error	Causa del error	Causa del error	Causa del error	Causa del error	Causa del error	Causa del error
Trámites		Análisis del código fuente	Análisis del código fuente	Análisis del código fuente	Análisis del código fuente	Análisis del código fuente	Análisis del código fuente	Análisis del código fuente	Análisis del código fuente	Análisis del código fuente	Análisis del código fuente	Análisis del código fuente
Métricas		Análisis de la documentación externa	Análisis de la documentación externa	Análisis de la documentación externa	Análisis de la documentación externa	Análisis de la documentación externa	Análisis de la documentación externa	Análisis de la documentación externa	Análisis de la documentación externa	Análisis de la documentación externa	Análisis de la documentación externa	Análisis de la documentación externa
Responsable												
Interfaces con otros procesos												

*Muy similares a las de la tabla anterior (correctivo), salvo la redacción del manual de usuario*

*3.1 = pg 29*  
*3.2 = pg 29*  
*3.3 = Redundante*  
*3.4 = 30 pg 29*  
*3.5 = 34 pg 29*

Tabla 7. Sinopsis de las actividades y tareas del evolutivo

Figure 3. A document with some notes taken during a joint meeting.



Modelo	MANTENIMIENTO CORRECTIVO RESUMEN MENSUAL		APL.	AÑO	MES	PAG.	
E010			TODAS	1998	marzo	1	
N.Max. peticiones mes (R1)	16	Tiempo de respuesta Muy Critico (R2mc)	0	Volumen del Codigo		1.412.362	
N.Max. peticiones semana (R3)	6	Tiempo de respuesta Critico (R2c)	5				
N.Max. pet. On Call mes (R1oc)	5	Tiempo de respuesta No Critico (R2nc)	12				
		Tiempo de respuesta On Call (R2oc)	5				
		Tiempo de respuesta Help Desk	0				
<b>Servicios Bancarios</b>							
<b>ANOMALIAS MUY CRITICAS</b>							
	N.Total Peticiones	N.Petic. Iniciadas	N.Petic. Finalizadas	T.Total Intervenc.	T.Medio Respuesta	Resp. > R2	N.Interv. 10.000 L.c.
Semana 9 (02-03-1998 - 08-03-1998)							
Semana 10 (09-03-1998 - 15-03-1998)							
Semana 11 (16-03-1998 - 22-03-1998)							
Semana 12 (23-03-1998 - 29-03-1998)							
Total del Mes							
<b>ANOMALIAS CRITICAS</b>							
	N.Total Peticiones	N.Petic. Iniciadas	N.Petic. Finalizadas	T.Total Intervenc.	T.Medio Respuesta	Resp. > R2	N.Interv. 10.000 L.c.
Semana 9 (02-03-1998 - 08-03-1998)	1	1	1	2:00	2:00		0,00708
Semana 10 (09-03-1998 - 15-03-1998)							
Semana 11 (16-03-1998 - 22-03-1998)	1	1	1	2:00	2:00		0,00708
Semana 12 (23-03-1998 - 29-03-1998)	1	1	1	4:00	4:00		0,00708
Total del Mes	3	3	3	8:00	2:40		0,02124
<b>ANOMALIAS NO CRITICAS</b>							
	N.Total Peticiones	N.Petic. Iniciadas	N.Petic. Finalizadas	T.Total Intervenc.	T.Medio Respuesta	Resp. > R2	N.Interv. 10.000 L.c.
Semana 9 (02-03-1998 - 08-03-1998)	1	1	1	2:00	2:00		0,00708
Semana 10 (09-03-1998 - 15-03-1998)	1	1	1	3:00	3:00		0,00708
Semana 11 (16-03-1998 - 22-03-1998)							
Semana 12 (23-03-1998 - 29-03-1998)							
Total del Mes	2	2	2	5:00	2:30		0,01416
<b>ANOMALIAS ON CALL</b>							
	N.Total Peticiones	N.Petic. Iniciadas	N.Petic. Finalizadas	T.Total Intervenc.	T.Medio Respuesta	Resp. > R2	N.Interv. 10.000 L.c.
Semana 9 (02-03-1998 - 08-03-1998)	4	4	4	8:00	2:00		0,02832
Semana 10 (09-03-1998 - 15-03-1998)							
Semana 11 (16-03-1998 - 22-03-1998)	11	11	11	30:00	2:44		0,07788
Semana 12 (23-03-1998 - 29-03-1998)	3	3	3	2:30	0:50		0,02124
Total del Mes	18	18	18	40:30	2:15		0,12745
<b>ANOMALIAS HELP DESK</b>							
	N.Total Peticiones	N.Petic. Iniciadas	N.Petic. Finalizadas	T.Total Intervenc.	T.Medio Respuesta	Resp. > R2	N.Interv. 10.000 L.c.
Semana 9 (02-03-1998 - 08-03-1998)							
Semana 10 (09-03-1998 - 15-03-1998)							
Semana 11 (16-03-1998 - 22-03-1998)							
Semana 12 (23-03-1998 - 29-03-1998)							
Total del Mes							

Figure 4. A monthly report of corrective maintenance.

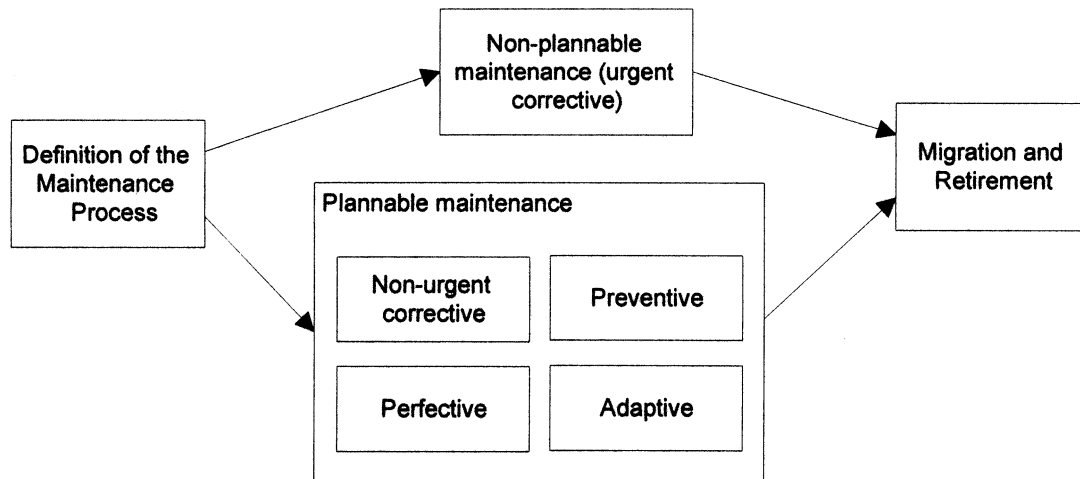


Figure 5. The structure of the process model after refinements.

activities to the methodology (explained in the next section) evidenced a lack of adaptation, since it is especially important to keep a record of corrective modification requests (there are several related Service Level Agreements covenanted between the Maintenance Organization and the Customer, see Section 4.5).

Such a distinction of types and their actual application also accelerated the Researcher's work, allowing us, for example, to reuse some tasks in different types of maintenance. As Table II shows, some tasks can be completely shared between different types of maintenance (such as the P1.3, 'Selecting alternative', which is common to both the non-urgent corrective and the perfective); others are partially shared (such as the P1.1 task, which defines special outputs for the adaptive maintenance, or P1.2, which is only applicable to non-urgent corrective and perfective, but with different and exclusive output products depending on the type) and others belong to just one type of maintenance (not shown in the table).

Until this moment, both the Researcher and the Critical Reference Group had only focused on the development of the 'Execution of the intervention' node of Figure 2; however, while the technical guides were being applied, the Researcher could dedicate most of their effort to the study of the 'Definition of the maintenance process' node.

#### 4.4. Incorporation of other activities

While the Critical Reference Group was applying the successive technical guides for urgent and non-urgent corrective, perfective, preventive and adaptive maintenance types, the Researcher continued advancing in the completion of the whole process model (remember we have only analyzed the middle node of Figure 2), which is the basis for constructing the whole methodology (see Section 3). As a

Table II. A fragment of the detailed structure of the plannable maintenance type.

		Activity		Intervention and tests
		Modification request analysis		
		Non-urgent corrective, preventive and adaptive	Non-urgent corrective and preventive	Non-urgent corrective and perfective
Tasks	P1.1	MR assessment	P1.2	P1.3
Inputs		Software product in operation Modification request	Documenting possible solutions Software product in operation MR in the waiting queue	Selecting alternative Software product in operation Implementation alternatives (DOC10)
Outputs		MR in the waiting queue Intervention calendar	CP Error diagnosis and possible solutions (DOC9) Implementation alternatives (DOC10) Product measures (DOC16a)	Selected alternative (full DOC9)
Techniques	A	Schedule estimation Resources disposability	P	Query to the historical DB
Metrics		Portfolio analysis Project management Time dedicated to the task	List of software elements and properties to improve (DOC12) Product measures (DOC16a) Source code analysis Project documentation analysis Time dedicated to the task Number of affected FPs Error origin and cause	Time dedicated to the task
Responsible		Maintainer	Maintainer	Maintainer
Int. processes			Quality assurance	

## Definitions:

MR = Modification Request

CP = Non-urgent corrective and perfective

P = preventive

DB = database

Int. processes = interfaces with other processes

result, we assumed the ‘process implementation’ and the ‘maintenance review/acceptance’ activities of ISO/IEC 12207 (almost as they appear in the Standard, although completing them with inputs, outputs, etc.) to compose the ‘Definition of the maintenance process’ node identified in Figures 2 and 5, which is comprised of two sets of tasks.

- The first set is executed only once, at the beginning of the maintenance project, and its goal is to prepare the future change process, such as to set the rules for accepting or rejecting modification requests or the implementation of the configuration management process. Tasks from the 12207 Standard were completed, for example, by including the documents that must be produced in each task (such as the technical summary of the software to be maintained, whose template was already defined and used by Atos ODS).
- The second set is executed once for each modification request received, and includes the reception of the request, the decision on its corresponding maintenance type (or its rejection), etc.

For the definition of this set of activities and tasks, we only had contact with the managers of the Maintenance and Outsourcing Unit, since neither programmers nor project managers are in charge of signing contracts, deciding on the acceptance of requests, etc.

In these same meetings, and also by analyzing the available data about projects reported by MANUTEN CONDUCT (such as time, product and process metrics or percentages of interventions of each maintenance type), the Critical Reference Group asked for the Researcher to extend the methodology to give support to outsourcing. As we stated in Section 1, Atos ODS supplies maintenance services to both large private and public companies. This kind of commercial relationship should be taken into account in the methodology.

Consideration of outsourcing entails the addition of new activities for the establishment and the ending of the outsourcing relationship; from a high level of abstraction, this just implies the addition of a new, initial node to the graph of Figure 5, as well as a new final one, resulting in the process model shown in Figure 6.

Mantema is usable with or without the existence of outsourcing. When there is outsourcing, the following three tasks of the ‘Initial activities of outsourcing’ node must be carried out; when there is not, they can be totally or partially omitted:

1. start-up and information collection;
2. preparing the maintenance contract;
3. contract.

After the execution of these tasks there is a written, formal contract between the supplier and the customer organization to provide and respectively receive the service; however, the actual provision of the maintenance service will not start until the execution of the ‘Knowledge acquisition’ task, that was placed in the second node of Figure 6 (‘Definition of the maintenance process’). This task is used by people in charge of the future maintenance interventions to learn how the software operates, working together with the current maintenance team during a period of about 1 or 2 months. At the end of this period, the service provider has enough knowledge to start the maintenance work, as well as updated documentation on the software, including audit documents that are delivered to the customer (thus avoiding the impression of not having worked).

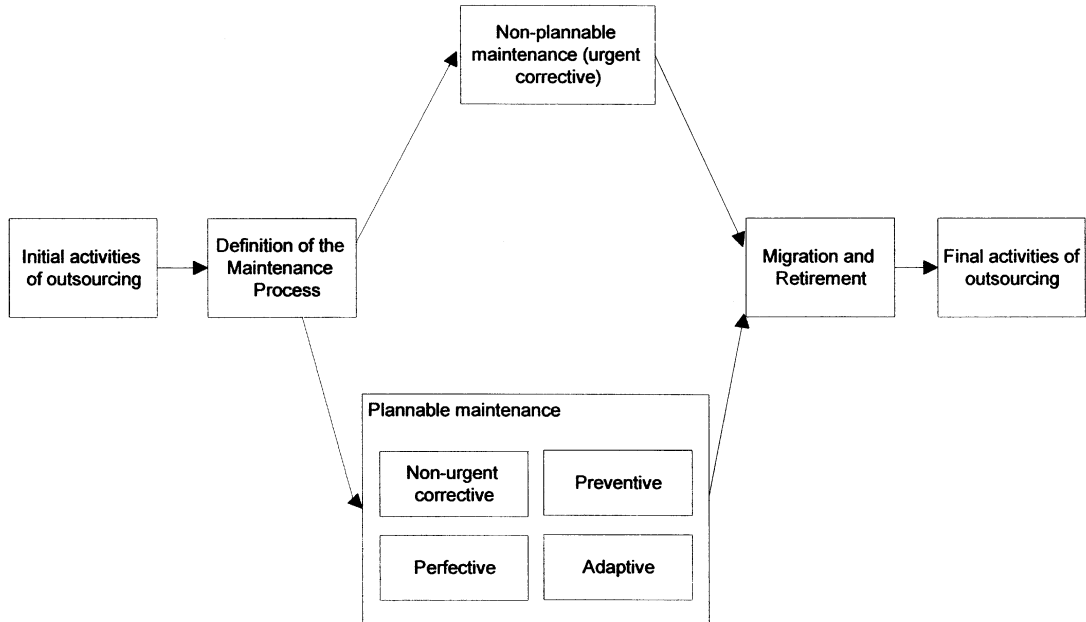


Figure 6. The final structure of the process model.

The last node of Figure 6 has one activity with three tasks, which must be carried out at the end of the outsourcing relationship:

1. delivering an inventory and documentation;
2. training new maintenance personnel (a task that can be considered as the opposite to that explained in the previous paragraph);
3. definite end of service.

A consequence of the introduction of outsourcing activities was the refinement of the three organizations we had identified at the beginning: it is clear that not all the people in the Maintenance Organization should be in charge of, for example, signing a maintenance contract. Moreover, not all of them are capable of adding a field to a table of the database. Therefore, a clearer identification of the people involved in the project was required. For this purpose, the Researcher studied some related literature, and made a proposal distinguishing several profiles in each organization. Later, in a joint meeting with Atos ODS staff, the proposal was tailored to their actual needs and situation [24].

#### 4.5. Completing the elements of the methodology

In spite of having focused our attention mainly on building the process model, a number of works were being developed in parallel. Their main aim was to propose elements in order to obtain a whole maintenance methodology.



According to Section 3, some of these elements are techniques, understood as practices that could help in the execution of some tasks. Some techniques found in the literature were added to fill in blank boxes of the process model (i.e. innovative techniques for reengineering or reverse engineering), but also some others were proposed to cover some deficiencies.

1. During the study of the activities required for outsourcing, a technique to estimate the risks associated with a maintenance project was developed [25]. This technique uses the set of situational factors of Euromethod for the acquisition of software systems [26], but is adapted to the maintenance environment and to the service supplier's point of view.
2. In spite of the 'non-plannable' name selected for the urgent-corrective maintenance, a technique to allocate and schedule resources for future urgent-corrective maintenance projects was proposed and empirically validated [27].

By using both techniques, the service supplier can get an idea of how many resources should be scheduled for a project or, even, for a period within a project. With these data, both parties can negotiate the project budget and consider other kinds of related indicators that were also defined and incorporated in the methodology:

1. maximum time for fixing errors due to urgent/non-urgent corrective modification requests;
2. maximum number of assumable urgent/non-urgent corrective anomalies per month;
3. maximum weekly deviation (i.e. maximum number of modification requests that can be received in a week);
4. sanction to be paid for each hour of delay in correcting urgent-corrective modification requests.

Other interesting commitments can be offered to the customer by the supplier service, such as decreasing the number of blocking errors after the contracted period, decreasing mean cyclomatic complexity, etc. To offer and guarantee the fulfilling of this commitment, the supplier must control the program evolution: Mantema proposes to collect a number of metrics in almost all its tasks, such as the number of lines of code added, changed or deleted, or the cyclomatic complexity of each routine before and after the intervention, etc. Moreover, as databases today are very important in information systems maintenance, although the research and usage of database metrics have been neglected, we have proposed and validated a wide set of metrics for several types of databases: relational and object-relational [28–30].

In many cases, these metrics have been both empirically and formally validated, and then incorporated into the methodology. Together with the input and output products, techniques, etc., every task includes a set of metrics that must be collected during its execution.

The results so briefly exposed in this section are not a trivial result or a consequence of the research, but the confirmation of Dick's opinion [31] that Action Research is 'a family of methodologies, each of which pursues action and research'. In fact, parallel to the cyclic Action Research process, other research methods were used for developing other elements of the methodology that were being incorporated in Mantema as they were finished.

On the other hand, the collection of metrics and the application of the methodology without automatic support were shown to be a difficult task by the final users of Mantema, which had already been transmitted to us on various occasions. To assist these issues, we developed Mantool, an automatic tool which allows the management of the software maintenance process according to the Mantema methodology [32].

Another aspect that also had been highlighted by the Critical Reference Group members was the lack of definition of the contents of some of the documents whose filling in was proposed in Mantema. As a response to this, we built a set of templates for all the possible documents that could be generated during the maintenance process. Mantool also incorporates these templates to collect data from users.

#### 4.6. Mantema as a whole

Every node of Figure 6 can be looked at in detail, thus obtaining a table in the style of Table II containing a rigorous descriptive structure. In these tables, the basic information required to perform every task is summarized, but methodology users can find more information in Mantema's technical manual. This, together with the automatic support offered by Mantool, makes the maintainer's work much easier.

### 5. CONCLUSIONS AND FUTURE WORK

In this paper we have presented the way in which we have used Action Research to build a methodology for software maintenance. From the Researcher's point of view, the simple fact of the experience itself has been extremely positive, since we have obtained good results and have explored and learnt in depth another very interesting approach to research. We are in agreement, therefore, with the idea of [5] that qualitative insight has a great importance in understanding real situations (which includes written reports, oral conversations and document analysis or observations, all complemented by research bibliography).

On the other hand, we have verified the perfect fit that Action Research offers to more classical research methods (i.e. quantitative ones). Returning to the terminology used in the rest of the article, the cyclical 'process model' of research allows the joint utilization of other vertical methods.

It is also important to highlight that the 'participatory' variant of Action Research has made the Critical Reference Group a very active participant, more so as the project advanced. In this manner, the Researcher is not seen as a mere consultant or auditor, but the Critical Reference Group feels itself as co-responsible for both the project development as well as the final results.

In spite of the positive experience obtained with Action Research, it would have been desirable to have more formal mechanisms for collecting data and opinions: obviously, the chain of conversations previous to our meetings with the Critical Reference Group introduced noise and information loss. However, we had to take this risk due to budget and time constraints (remember that the company was tackling the Y2K and euro effects).

The benefits that Mantema has provided to the Critical Reference Group are also favourable. In fact, many of the drawbacks that the previous maintenance management method had have disappeared. Mantema allows the control of the process to be maintained through the continuous measurement of products and of the process itself; it facilitates the interchange of personnel between projects thanks to the continuously-updated documentation and to the identification of people in charge of tasks, which in turn decreases the costs and the problems of people leaving.

Mantema has contributed to Atos ODS obtaining the ISO 9000 certification in the maintenance service rendered to customers. The initial framework used to build Mantema (the ISO/IEC 12207 IS) allows its future expansion and the covering of some deficiencies, such as its relationships with

other life-cycle processes: the maintenance process defined in Mantema has several interfaces with supporting and organizational processes of the 12207 Standard (for example, with the documentation process). As some of them have been left out of Mantema, we are and have been working on the integration of some other processes which had not previously been considered as an important part of the maintenance. So, for example, the audit (a supporting process) of the maintenance process was analyzed in [33], whereas the incorporation of improvement activities (another supporting process) to Mantema was studied in [34]. In connection with this issue, we are involved in new projects both with Atos ODS as well as with other companies applying, of course, Action Research.

#### APPENDIX A. THE MAIN MILESTONES OF THE PROJECT

Year	Event	Participants	
		Atos	Alarcos
1997	Mantema project signature	✓	✓
	Problem exposition	✓	✓
	Study and recollection of bibliography		✓
	Building an initial structure of the methodology		✓
	Technical meetings for refining such a structure	✓	✓
	Refinement of the initial structure		✓
	New technical meetings	✓	✓
	Key milestone: proposal of separate treatment for every maintenance team		✓
1998	Definition of a technical guide for corrective maintenance		✓
	Application of such 'mini methodology' to real projects	✓	
	Refinement of technical guides for the rest of maintenance types		✓
	Comment on corrective maintenance: a new division, urgent and non-urgent corrective	✓	
	Presentation of a methodology for perfective maintenance		✓
	Study and short comments of the methodology for perfective maintenance	✓	
1999	Presentation of Mantema version 1: five technical guides for five types of maintenance (urgent and non-urgent corrective, perfective, preventive and adaptive)		✓
	Application of Mantema version 1 to real projects	✓	
	Study of Mantema version 1 for the correction of defects and addition of characteristics (outsourcing, etc.)		✓
	Search in the literature and study of techniques for helping in certain tasks of the maintenance process		✓
	Proposal of techniques for planning resources		✓
	Study of techniques for estimating risks in outsourcing maintenance projects		✓
	Generation of templates for maintenance documents		✓
	Report of faults during the application of Mantema version 1	✓	

Year	Event	Participants	
		Atos	Alarcos
	Report advising on the convenience of setting apart urgent corrective, and of grouping into just one type ( <i>non-plannable</i> maintenance) non-urgent corrective, perfective, preventive and adaptive ( <i>plannable</i> )	✓	
	Building of Mantema version 2		✓
	Final stage of Mantema project (I): application of Mantema version 2 to Atos ODS projects	✓	
2000/	Results of the application of risk techniques	✓	
2001	Refinement of techniques; validation with historical data		✓
	Final stage of Mantema project (II): incorporation into the methodology of the last comments of Atos ODS; compilation of Mantema version 3.0		✓

## APPENDIX B

The different technical guides elaborated for the five types of maintenance were used in several pilot projects of one of the most important customers of Atos ODS (one of the biggest Spanish banking corporations). These projects are all implemented in Cobol, Cobol-DB2, Cobol-CICS and Cobol-CICS-DB2:

- *Tax collection*. This application has 135 programs with 103.331 LOC, and its main functions are: tax collection for the Ministry of Finance, autonomous regions and city councils; information collection on tax refunds to be sent to the Ministry of Finance; and seizures of accounts under orders from the Ministry of Finance.
- *Receipt collection*. It has 157.281 lines of code in 196 programs. It manages the payment of the debits received via diskettes or via electronic interchange with other corporations.
- *Transfers*. It has 308 programs and 247.156 lines of code. This application manages transfers between accounts of the same or different entities, both periodic or sporadic.
- *Pro-active risk*. This project analyses all the available information of the clientele in order to automate the granting of credit. It was developed in 1994 and 1995 and runs on an IBM 9000-982 with MVS-TSO. It has 576 programs, 235 with access to DB2 and 94 on-line transactions, which execute 227 different programs.

## ACKNOWLEDGEMENTS

The authors wish to thank the anonymous reviewers whose suggestions have considerably improved the paper. This work was partially supported by the MANTIS project (grant number CICYT 1997-1608 TIC, Ministerio de Ciencia y Tecnología) and the MPM project (PROFIT 070000-2000-307).

## REFERENCES

1. Pigoski TM. *Practical Software Maintenance. Best Practices for Managing your Investment*. Wiley: New York, 1997.
2. Singer J. Practices of software maintenance. *Proceedings International Conference on Software Maintenance*. IEEE Computer Society Press: Los Alamitos, CA, 1998; 139–245.
3. Van Bon J. Sourcing. *World Class IT Service Management Guide*, Van Bon J (ed.). ten Hagem & Stam Publishers: The Hague, The Netherlands, 2000.
4. Schach SR, Tomer A. A maintenance-oriented approach to software construction. *Journal of Software Maintenance: Research and Practice* 2000; **12**(1):24–45.
5. Basili V, Briand L, Condon S, Kim Y, Melo W, Valett JD. Understanding and predicting the process of software maintenance releases. *Proceedings International Conference on Software Engineering*. IEEE Computer Society Press: Los Alamitos, CA, 1996; 464–474.
6. Niessink F, van Vliet H. Software maintenance from a service perspective. *Journal of Software Maintenance: Research and Practice* 2000; **12**(2):103–120.
7. Polo M, Piattini M, Ruiz F. Managing the software maintenance process. *World Class IT Service Management Guide*. Van Bon J (ed.). ten Hagem & Stam Publishers: The Hague, The Netherlands, 2000.
8. French WL, Bell CH. *Organizational Development: Behavioral Science Interventions for Organization Improvement*. Prentice-Hall: London, 1996.
9. Avison D, Lau F, Myers M, Nielsen A. Action research. *Communications of the ACM* 1999; **42**(1):94–97.
10. Seaman CB. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering* 1999; **25**(4):557–572.
11. Kock N, Lau F. Information systems action research: Serving two demanding masters. *Information Technology and People* 2001; **14**(1):6–11.
12. Wadsworth Y. What is Participatory Action Research? Action Research International. <http://www.scu.edu.au/schools/sawd/ari/ari-wadsworth.html> [12 June 2002].
13. Padak N, Padak G. Guidelines for Planning Action Research Projects. <http://archon.educ.kent.edu/Oasis/Pubs/0200-08.html> [12 June 2002].
14. Graham I, Henderson-Sellers B, Younessi H. *The OPEN Process Specification*. ACM Press and Addison-Wesley: Essex, UK, 1997.
15. European Commission. Employment in Europe 2001—Recent trends and prospects. Office for Official Publications of the European Communities, Luxembourg. [http://europa.eu.int/comm/employment\\_social/empl/esf/docs/empleurope2001\\_en.pdf](http://europa.eu.int/comm/employment_social/empl/esf/docs/empleurope2001_en.pdf) [12 June 2002].
16. IEEE. *Standard for Software Maintenance*, IEEE Std. 1219-1992. The Institute of Electrical and Electronics Engineers, Inc.: New York, 1992.
17. ISO/IEC. *Information Technology-Software Life Cycle Processes*, ISO/IEC 12207. Swiss, 1995.
18. Mazza C, Fairlough J, Melton B, de Pablo D, Scheffer A, Stevens R. *Software Engineering Guides*. Prentice-Hall: London, 1996.
19. Radatz J, Olson M, Campbell S. MIL-STD-498. *Crosstalk, the Journal of Defense Software Engineering* 1995; **8**(2):2–5. <http://www.stsc.hill.af.mil/crosstalk/1995/feb/milstd.asp> [12 June 2002].
20. Ministerio de Administraciones Públicas. *Métrica 3: guía de referencia*. Tecnos: Madrid, 2001. <http://www.map.es/csi/metrica3> [12 February 2002].
21. Polo M, Piattini M, Ruiz F, Calero C. Mantema: A complete rigorous methodology for supporting maintenance based on the ISO/IEC 12207 standard. *Proceedings Third European Conference on Software Maintenance and Reengineering*. IEEE Computer Society Press: Los Alamitos, CA, 1999; 205–210.
22. Pressman RS. *Software Engineering: a Practitioner's Approach* (3rd edn). McGraw-Hill, 1993.
23. Piattini M, Villaba J, Ruiz F, Bastanchury T, Polo M, Martínez MA, Nistal C. *Mantenimiento del Software. Modelos, técnicas y métodos para la gestión del cambio*. Ra-Ma: Madrid, 2000.
24. Polo M, Piattini M, Ruiz F, Calero C. Roles in the maintenance process. *ACM Software Engineering Notes* 1999; **24**(4):84–86.
25. Polo M, Piattini M. Cuestionario para la identificación de riesgos en proyectos de mantenimiento. *Actas de las V Jornadas de Ingeniería del Software y Bases de Datos*, Universidad de Extremadura, Cáceres, Spain, 2000; 30–35.
26. Euromethod project. Euromethod ver 1. <http://projekte.fast.de/Euromethod/> [12 June 2002].
27. Polo M, Piattini M, Ruiz F. Planning the non-plannable maintenance. *Project Control: the Human Factor, Proceedings of the Combined 11th European Software Control and Metrics Conference and the 3rd SCOPE Conference on Software Product Quality (ESCOM-SCOPE 2000)*. Shaker Publishing: Maastricht, The Netherlands, 2000; 49–57.
28. Piattini M, Calero C, Ruiz F. Table oriented metrics for relational databases. *Software Quality Journal* 2000; **9**:79–97.
29. Piattini M, Calero C, Sarahoui HA, Luonis H. An empirical study with object-relational databases metrics. *4th International Workshop on Quantitative Approaches in Object-Oriented Software Engineering (in ECOOP'2000)*, Centre de recherche informatique de Montreal, Montreal, Québec, 2000; 39–46.

- 
30. Díaz Ó, Piattini M, Calero C. Measuring triggering interaction complexity in active databases. *Information Systems Journal* 2001; **26**(1):15–34.
  31. Dick B. Postgraduate programs using action research. Action learning, action research and process management: theory, practice, praxis, Zuber-Skerritt (ed.), Faculty of Education, Griffith University, Brisbane, Australia, 2000.
  32. Polo M, Piattini M, Ruiz F. Mantool: a tool for supporting the software maintenance process. *Journal of Software Maintenance and Evolution: Research and Practice* 2001; **13**(1):77–95.
  33. Ruiz F, Piattini M, Polo M, Calero C. Audit of software maintenance process. *Auditing Information Systems*, Piattini (ed.). Idea Group Publishing: Hershey, PA, 2001.
  34. Polo M, Piattini M, Ruiz F, Jiménez M. Assessment of maintenance maturity in IT departments of public entities: Two case studies. *Proceedings of the 3rd International Conference on Product Focused Software Process Improvement (Lecture Notes in Computer Science*, vol. 2188). Springer: Berlin, 2001; 86–97.