Information Technology and Management 3, 247–269, 2002

Integrating Outsourcing in the Maintenance Process

MACARIO POLO, MARIO PIATTINI and FRANCISCO RUIZ
{macario.polo, mario.piattini, francisco.ruizg}@uclm.es
Grupo Alarcos Escuela Superior de Informática, Universidad de Castilla-La Mancha, Ronda de Calatrava, 5, E13071-Ciudad Real (España)

Abstract. Outsourcing of software life cycle activities is a growing business area in many sectors influenced by Information Technologies. This fact, coupled with the usual lack of planning and high costs of software maintenance, may invite many organizations to outsource this important process of the software life cycle. Such outsourcing should be relied to a "technological associate" who can carry out this process using an adequate methodological foundation. In this paper we present the outsourcing strategy that we have integrated in MANTEMA, a methodology for software maintenance developed by our university and Atos ODS, a multinational organization which provides software maintenance services to third-party organizations.

Keywords: software maintenance, outsourcing, risks management, maintenance planning

1. Introduction

Outsourcing of software life cycle activities is a growing business area in many sectors influenced by Information Technologies (IT) [30]. Even the most unexpected organizations, such as the US Department of Defense, have decided to outsource significant portions of their Information Systems [4]: in fact, according to [12], 40% of the largest companies in the United States have outsourced at least one of the major parts of their operations.

One of the possible reasons of this "epidemic outbreak" of outsourcing is the uncontrolled expansion undergone by the IT departments some years ago (earl reports on organizations that have spent more than 50% of total expenditure on IT [9]), as well as a lack of planning and little integration into the strategy of the organization. This meant an increase in the percentage of investments on IT that, in some cases, threatened the companies' results.

Among the activities that comprise the software life cycle, maintenance is the most costly. Some authors place these costs between 67 and 90% of total life cycle costs [6,22].

Besides the lack of a rigorous definition of the maintenance process [2,22], the absence of scheduling of the change interventions is a determinant point regarding these high costs. In fact, whereas the production of new developments does not affect the organizations' daily life, maintenance is intimately bound to routine problems. The reason for this divergence is that, in new developments, the stages, meetings, follow-ups, etc. are previously scheduled and are backed with a high degree of motivation in the staff.

However, this situation does not exist in maintenance, where unneeded moments of edginess and pressures are multiplied, increasing staff dedication to these tasks beyond the desired level.

Therefore, it seems necessary to control the maintenance process by knowing it, and not by attempting to adapt to it structures and organizations that are planned for new developments. This necessity could be provided by a third-party organization, an information systems maintenance supplier that, playing the role of “technological associate”, should be a specialist in the application of best software practices and methodologies to this costly process.

One of these maintenance methodologies is MANTEMA [23,26], developed jointly by our university and Atos ODS, one of the most important european consultants on software maintenance outsourcing. This organization is applying this methodology in an increasing number of its projects, mainly related with big banking, industrial and telecommunication organizations.

MANTEMA was built adapting the ISO/IEC 12207 Maintenance Process applying the Tailoring Process of this same International Standard [26]. Recently, Atos ODS was awarded the ISO 9000 certification as a Maintenance Services provider, where the use of MANTEMA is an essential factor.

In this article we present how MANTEMA integrates outsourcing activities in the maintenance process. MANTEMA was designed with the supplier service organization in mind and because of this, it does not propose special techniques nor methods for potential customers. Activities and risks associated with outsourcing from the customer’s point of view have been widely studied by other authors (see section 2).

Obviously, although this vision is adapted to providing maintenance services, it can be also easily adopted in other environments and contexts, such as software development or network solutions, which are two of the most outsourced areas in IT [36].

The article is organized as follows: section 2 briefly describes some of the pros and cons of outsourcing. Section 3, has a short presentation of the structure of MANTEMA methodology. Section 4 explains and details the activities of MANTEMA which can serve as guidelines for the establishment and finalization of outsourcing relationships. Section 5 holds our conclusions and future lines of work.

2. Outsourcing the information system

Outsourcing relationships always entail some dangers that may affect both contracting parts. Before accepting a maintenance project, outsourcing suppliers must carefully analyze the proposal and the software product of the potential customer (the party that wants to acquire the maintenance service) in order to prepare the maintenance contract, budget, resource allocation, etc., and also, to decide whether to accept or, if the importance of the customer allows it, to reject the project. In the same way, the customer organization should evaluate the problems related with the transfer of its Information System (IS) to an outsider.

Benefits of outsourcing all or a part of an organization's IS are [7,16]:

- *Employees center their attention on the core business*, allowing the company to concentrate on strategic IS issues.
- *Releasing resources for strategic developments*, which further allows the achievement of greater flexibility in teams. The use of the most qualified resources for activities highly bonded to business goals, such as the management of new systems or the planning and follow-up of the outsourcing relationship, becoming the intermediate step between the maintenance supplier and the customer.
- *Decreasing costs*, through:
  - Savings in personnel training and management, as these costs now move to the supplier.
  - Transformation from fixed to variable costs: for example, before outsourcing, nobody in the organization knows how much money will be spent correcting software errors, whereas after it there will be a fixed price, submitted by the supplier for corrective maintenance during a period. Knowing and fixing costs before the service is given, allows for a more exact profit/cost analysis.
- *Increasing productivity*, through:
  - Elimination of maintenance interferences on resources dedicated to new developments.
  - Benefits produced because maintenance is now carried out by a technological associate whose essential competence is focused on this activity; that is, a specialist who, in order to maintain itself in the market, must invest in Research and Development for its own profit.
- Outsourcing of the maintenance process also allows:
  - A greater control over the process, since maintenance interventions will be planned.
  - Decreasing corrective maintenance through a commitment during the outsourcing period. The supplier acquires the commitment of decreasing the required level of corrective maintenance (i.e., software errors) at the end of contracted period. The supplier reaches this goal through continuous improvement of software quality, which is attained by means of continuous preventive maintenance.

However, outsourcing also involves a set of drawbacks which must be taken into account by customer:

- *loss of control* and *loss of a learning source*, because an internal activity is externalized;
- *loss of knowledge of the software*;
- *possible dependencies on the supplier*;

– *variations in the quality of the product given to the customer*;
– *problems among personnel*, since they lose their functions.

Of course, the magnitude of these benefits and risks depend on the qualifications of the supplier. This is the principal reason for applying methodological foundations when software services are supplied.

3. Brief presentation of MANTEMA structure

Following recommendations made in literature [13,14,28], MANTEMA distinguishes several types of maintenance interventions (urgent corrective, non-urgent corrective, perfective, preventive and adaptive) and defines each one meticulously.

However, the initial application of the methodology to real projects evidenced that this classification was not very practical and a conceptual grouping of the last four types was made into just one, corresponding to *plannable maintenance* (although each one really keeps some characteristics which makes it different from the rest). In this sense, and in order to maintain the same notation, *non-plannable maintenance* has been grouped under urgent-corrective.

On the other hand, and due to the imperative necessity of considering the establishment and ending of the outsourcing relationship as a part of the maintenance process, we add a set of activities to be executed at the beginning and at the end of our process: the former (node labeled "Common initial activities and tasks" in figure 1) are focused on the preparation of the outsourcing relationship and on the preparation of the process

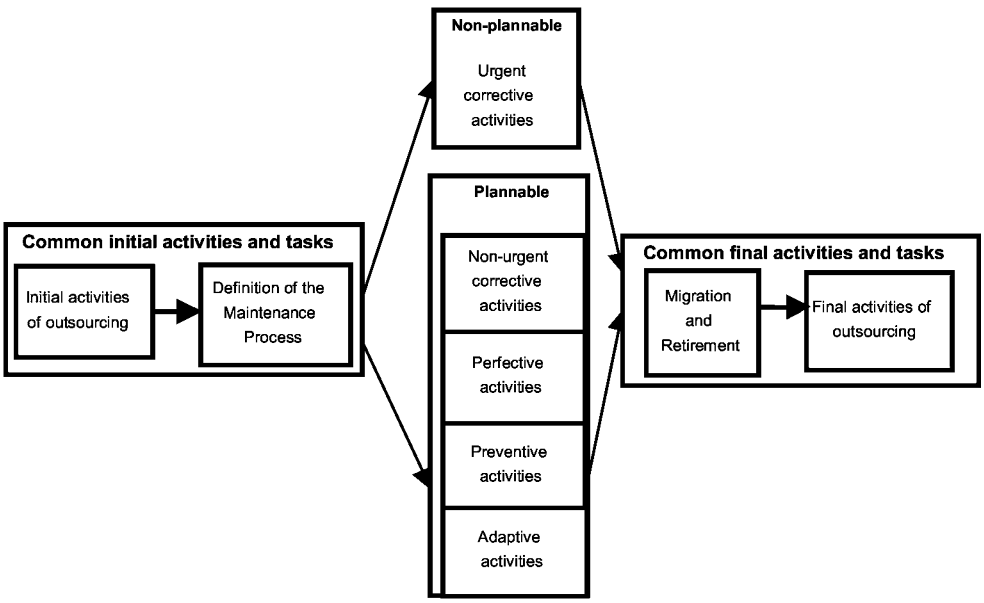

Figure 1. Consideration of outsourcing and different types of maintenance.

itself; the latter (subnode labeled "Final activities of outsourcing"), on the ending of the relationship.

Every node of the process model is composed of activities, and every activity by a set of tasks. Tasks are defined by providing their input and output products, techniques, metrics, people responsible and interfaces with other processes. Table 1 shows some of these elements for first tasks in MANTEMA. With this structure, MANTEMA is a full methodology in the sense given by Graham, Henderson-Sellers and Younessi, for whom a methodology must provide, at least, a process model, techniques, guides for the project management (including roles and team structure), deliverables, metrics and automatic tools for supporting the methodology [11]. This article deals with the first three issues.

In section 4 we will use different terms to make reference to the responsible people of every task. These people are really profiles of the organizations involved in the maintenance process; in order to uniquely identify each organization and profile, they have been discussed and characterized in [25]. Table 2 is a brief reminder in order to assist reading this article.

4. Maintenance outsourcing support in MANTEMA

As we can see in figure 1, the establishment of an outsourcing relationship is made in the first subnode of the *Common initial activities and tasks*, which is also represented in the first row of table 1. The goal of the first activity in table 1 (*Initial study*) is to set out the foundations to establish the outsourcing relationship. Once the outsourcing contract is signed, the maintenance process is prepared and implemented in the second activity (*Process implementation*), although software modifications still remain under the control of the acquirer's organization. The supplier organization begins its effective work as maintainer from the third activity onward.

The followings epigraphs analyze these activities and their tasks in more depth.

4.1. Activity 0: Initial study

This activity is executed when the customer organization has contacted the maintainer, to acquire the maintenance service for all or a part of its Information System. During this activity, the maintainer does not have the responsibility of software modification interventions. This activity consists of the following tasks:

4.1.1. Task 0.1: Beginning and information recollection

This task is triggered as a response to a "Maintenance Service Request" from the Customer. The maintenance team, managed by the head of maintenance (and in collaboration with the system organization profile), fills in a document called *Initial Questionnaire* (figure 2) which details some issues of the software to be maintained.

Table 1
Initial activities and tasks.

	Common initial activities and tasks										
	Initial study			Process implamentation						Study of the modification request	
Tasks	10.1 Beginning and information recollection	10.2 Preparing maintenance proposal	10.3 Contract	11.0 Planning customer/supplier relations	11.1 Knowledge acquisition	11.2 Developing plans	11.3 Defining modification request procedures	11.4 Implementing Configuration Management process	11.5 Preparing test environments	12.1 Reception of the modification request	12.2 Decision about the type of intervention
Inputs	Request of Maintenance service	Questionnaire interviews	Maintenance proposal	Maintenance contract	Software product in operation	Software product in an operation	Maintenance plan	Software product in operation Maintenance plan	Software product in operation	Modification request	Received modification request
Outputs	Initial questionnaire (DOC1)	Maintenance proposal	Maintenance contract	Customer/ supplier relations planning (responsibilities, meetings calendar, etc.)	Software product in operation yet new documents about such software	Maintenance plan	Document models and forms regulatory ways	Configuration Management Process	Copies of the software elements in operation	Received modification request	Report about type of maintenance, criticise, etc. Decision about the type of maintenance to apply
Responsible	Customer Maintainer	Maintainer	Customer Maintainer	Customer Maintainer	Maintainer	Maintainer	Maintainer	Maintainer	Maintainer	Customer Maintainer	Maintainer
Interfaces with other processes				Tailoring				Configuration Management	Quality assurance		

A. Organization identification (financial data, personnel, etc.) and authorized speakers

B. Hardware and software environment

- B.1. Hardware
  - B.1.1. Mainframes (brand and model, operating system, etc.)
  - B.1.2. Secondary hardware
    - 1.2.1. Terminals
    - 1.2.2. Workstations (indicating operating system)
    - 1.2.3. Network type
    - 1.2.4. Connection type with the main hardware
- B.2. Software environment
  - B.2.1. Development environment
  - B.2.2. File system
  - B.2.3. Databases
  - B.2.4. JCL Batch

C. Development and maintenance organization

- C.1. Development methodologies and techniques
- C.2. Use of denomination and programming standards
- C.3. Quality management techniques and Standards
- C.4. Project management procedures
- C.5. Operation procedures
- C.6. Auditing procedures
- C.7. Problem resolution procedures
- C.8. Documentation procedures
- C.9. Other procedures o standards

D. Applications (for every one)

- D.1. Identification (name and date put into production)
- D.2. Organizational unit responsible/user
- D.3. Related applications
- D.4. Batch programs (languages, size, year of writing and number of interventions)
- D.5. On-line programs (languages, size, number of modules, year of writing and number of interventions)
- D.6. Reports (number, year of writing and number of interventions)
- D.7. Screens (number, year of writing and number of interventions)

Figure 2. Initial questionnaire.

Table 2
Organizations and profiles involved in the maintenance process.

Organization	Profile	Short description
Customer	Petitioner	promotes a Modification Request, defines the needed requirements for its implementation and informs to the Maintainer
	System organization	department that has a good knowledge of the system that will be maintained
	Help-desk	department which attends to users. It also reports to the *Petitioner* incidents sent by *users* to generate the Modification Request
Maintainer	Maintenance-request manager	decides whether the modification requests are accepted or rejected and what type of maintenance should be applied. He/She gives every Modification Request to the *Scheduler*
	Scheduler	plans the queue of accepted modification requests
	Maintenance team	group of people who implement the accepted modification request. They take modification requests from the queue, which is managed by the *Scheduler*
	Head of maintenance	prepares the maintenance stage. He/She also establishes the standards and procedures to be followed with the maintenance methodology used
User	User	makes use of the maintained software. He/She communicates the incidents to the *Help-desk*

1. Introduction
2. General results of the studies made of the analyzed applications
3. Service proposal:
   - 3.1) technical report defining: goals, bounds, bindings, responsibilities and contractual parameters;
   - 3.2) for every application inside the (future) maintenance contract, maintenance types and their corresponding Service Level Agreements must be set;
   - 3.3) proposal of contract (that will have the same format as the Final contract)
4. Economic offer

Figure 3. Contents of the maintenance proposal.

4.1.2. Task 0.2: Preparing the maintenance proposal

From data collected in the previous task, the head of maintenance prepares a Maintenance Proposal (figure 3) for the customer organization, which has the basics for the future contract. Obviously, the maintainer must make estimations regarding maintenance costs, taking the data collected in the Initial Questionnaire as the basis for calculations.

Some internal reports and studies must be made before presenting this document to the customer, specially related to economic and resources' issues. MANTEMA contributes to this aspect with several techniques:

- a method for identifying and estimating the areas of the software system with most risk (sections 4.1.2.1 and 4.1.2.2);
- a novel set of Service Level Agreements, which allow to, previously and objectively, know the quality levels of the service provided by the maintainer, as well as the ulterior control and monitorization of the process during the future modifications of the software (section 4.1.2.3);
- a method for estimating the quantity of human resources to be devoted to non-plannable maintenance (section 4.1.2.4).

4.1.2.1. Risk identification

There are many good works analyzing the problem of risks once the maintenance process has started (for example [32,35]), but there is a lack of guidelines to help managers identify and estimate the risks in the initial stages of projects. Cost estimation techniques might be considered an exception to this point; however, they do not consider risks in such estimations (although the weight assignation table in Albrecht's function-point could be considered as a simple form of estimation [1]) or, if they do, these are relied to an expert's judgment without providing more details [3,34].

General methods of risk management usually split up the process into a set of subprocesses [15,29] that may and must be iterative. As the goal of this article is limited to actions prior to the contract, we only analyze points 1 and 2 from the following:

1) risk identification;
2) risk quantification;
3) risk response development;
4) risk response control.

Within the scope of this article, we propose the maintainer use a risk checklist. The first version of this list was built from some of Euromethod's situational factors [10] and other characteristics of projects historically collected by Atos ODS's. Although our partner collected and evaluated this kind of data in almost all its projects, the truth is that they did not result very useful for any estimation. However, according to [10], situational factors are "those properties of the situation which generate risks and which should be taken into account in the design of the acquisition strategy"; therefore, as one situational factor may affect one or more risk factors, and each risk factor may affect on one or more business areas [18], it is important to measure and understand the relationship between situational factors and projects risks.

Before constructing version 1 of the checklist, we did a Principal Component Analysis of factors historically collected by Atos ODS, in order to detect groups of factors which measured the same project property. We found that several orthogonal groups

Table 3
Principal Components obtained.

CP	Eigenvalues		
	Total	% variance	% accum var.
1	11.831	32.864	32.864
2	8.167	22.686	55.549
3	6.238	17.329	72.878
4	4.453	12.370	85.249
5	3.378	9.383	94.632
6	1.456	4.046	98.678

of factors were providing the same information, although this was not being used. After this study, we incorporated some of the Euromethod's factors, obtaining the first version of the MANTEMA's checklist with 36 questions.

This first version was delivered to the people responsible of sixteen large projects, twelve of Atos ODS', integrally developed in Cobol, Cobol-CICS and Cobol-CICS-DB2, and four projects of an important public institution, developed in a 4GL. Each one has between 125.000 and 275.000 code lines. A new Principal Component Analysis was made on the data collected, which highlighted the six Principal Components shown in table 3, that accumulate more than 98% of the total variance. This produces a questionnaire that evaluates the 21 essential situational factors (an additional benefit of the reduction of the number of factors is the increase in ease and speed in answering the questionnaire).

Table 4 shows the final version of the checklist: the described building method is the reason why it contains only the situational factors of interest to the maintainer.

Situational factors are listed in the second column of table 4; each one belongs to the group indicated in the first. Every cell in the third column must be filled in with values from 1 to 5, according to what the user of the table feels about the entry in the second column (*Absolutely agree*, *Agree, Indifferent*, *Disagree* or *Absolutely disagree*, respectively). After this, the fourth column is completed with the mean values of its corresponding section, or with a weighted value, depending on the organization's experience.

4.1.2.2. Risk estimation

Once the previous table has been completed, one can advance to the following step, that is, the quantification of risks for the supplier organization. According to Euromethod, a risk "is the possibility of exposure to the adverse consequences of future events". The grouping of the situational factors according to their source (first column in table 4) will allow us to know the weakest and most dangerous points of the customer environment within our scope. Each *source* influences certain types of risks.

In table 5 we reproduce the most usual risks in maintenance projects. Every risk is quantified by moving values in the fourth column of table 4 (mean or weighted values of every source of situational factors) to the corresponding blank cells in

Table 4
Situational factors for maintenance outsourcing.

Factor	Subfactor	Value	Mean or weighted value
Existing specifications of the application	High quality of the design level of the system technique architecture High quality of the level design of the software IS requirements are available and clear IS requirements are stable		
Current team of development/ maintenance of the customer …	… has a high work capacity … has a high knowledge level of the *Maintainer* referred to the domain of the system[a] … has a high knowledge level of the *Maintainer* referred to the technical environment of the IS[a]		
Extern factors	Organization does not depend much on subcontractors		
IS factors	Business processes are simple Information is stable		
Critical factors of the application	There are no random critical moments There are no periodic tops which require incorporation of people or greater resources dedication (for example, at the end of the month) There is no dependence of the uer to know the problem		
Quality of the application	Denomination and programming standards have been widely used Quality of batch programs is high (depends on the number of programs, languages used and other quality factors, as lines of code, number of modules, age, number of interventions, cyclomatic complexity, coupling of the modules, etc.) Screens are simple and little influenced by maintenance interventions		
Organization factors	There is little dependence on organizational changes There is a low change rate in business processes		
Methodologies	Technology used in the project is available Methodologies were used during the development of the system Tools were used during the development of the system		

[a] In MANTEMA, after signing the outsourcing contract, there is a period under tutelage during which the outsourcing organization acts as a mere observer, observing how the maintenance organization works (up to then). The goal of this period is to obtain a faster knowledge of the software domain.

table 5. This correspondence between situational factors and risks has been obtained through a statistical analysis of data from the previously mentioned sixteen projects.

After this, the values of the rows in table 4 are added and divided by the corresponding number on its right. This number represents the maximum value possible for the risk factor, which is obtained by multiplying 5 (value provided when the user's answer in table 4 is *Absolutely disagree*, which corresponds to the worst possible value)

Table 5
Relationships between situational factors and risks.

		Sources of situational factors										
		Existing specifications of the application	Current team of development/maintenance of the customer	Extern factors	IS factors	Critical factors of the application	Quality of the application	Organization factors	Methodologies	Total sum	Risk value	
Risks	Uncertain or unfeasible requirements										/10	
	Uncertain interfaces to other systems										/5	
	Evolving requirements										/20	
	Unpredictable costs for the supplier organization										/20	
	Unpredictable costs for the project										/20	
	Delays in the delivery										/15	
	Poor quality of deliverables										/15	
	Increased costs of the project										/10	
	Integration problems										/10	
	Straining computer science capabilities										/5	
	Wrong or unfeasible information system										/20	
	IS-adaptation not accepted by actors										/5	
	Business implications of project failure										/10	

by the number of blank cells in every row. Obviously, if weighted values were used in table 4, the numbers used as denominators in table 5 should be reviewed.

4.1.2.3. Service Level Agreements

In order to offer a competitive service that, at the same time, is objectively measurable, the maintenance organization must commit itself to accomplish its services according to some indicators. Different “Service Level Agreements” for every contracted service (including types of maintenance) must be proposed and agreed upon in the contract.

Setting aside other usual indicators [8], MANTEMA defines the following, although this list can obviously be extended, if it is so required:

- NMR_{CA}, which is the *Number of Modification Requests related to Critic Anomalies* assumable per period (for example: 30 per month);
- NMR_{NCA}, which is the *Number of Modification Requests related to Non-Critic Anomalies* assumable per period;
- TRCA (*Time of Resolution of Critic Anomalies*): the maximum time that the maintenance organization may employ in fixing a critic anomaly (urgent-corrective) without being sanctioned;
- TRNCA (*Time of Resolution of Non-Critic Anomalies*): the maximum time to resolve non-critic modification requests (non-urgent corrective) without being sanctioned. Obviously, four different values for TRNCA may be defined, one for each type of plannable maintenance;
- PPC (*Progressive Preventive Commitments*): although interventions of preventive maintenance are typically not agreed upon, however a commitment of progressive and continuos preventive maintenance is covenanted (for example, the maintenance organization commits itself during the period to decrease the mean cyclomatic complexity of all the modified modules, or to reduce the number of errors).

The commitment of Service Level Agreements for the rest of plannable types of maintenance (specially perfective and adaptive) is not frequent and is not recommended, since it is difficult to anticipate time and effort for these kinds of interventions, which may imply big functionality additions. Interventions of these types should be studied individually.

These Service Level Agreements can also be used for *planning the non-plannable maintenance*, as we will see in the following section.

4.1.2.4. Planning the non-plannable maintenance

We have named "non-plannable" as urgent corrective, as it is precisely, the less plannable and more problematic type of maintenance, due to its lack of planning possibility. However, some authors have already modeled a typical life cycle for maintenance, including corrective maintenance [17], for instance observes how most requests of this type of maintenance are concentrated in the later moments of every software release. [2] and [33] have shown similar results. [5] have proposed a method (based upon the dynamic life model of May) for predicting the arrival of maintenance requests, including corrective maintenance. The ANGEL tool [34], which predicts effort using analogy, has been used by [19] in maintenance projects.

The technique we explain below may use any predictive model of maintenance request distribution, together with economical parameters of the project, for determining the quantity of resources dedicated to error corrections during a period, in such manner that the maintainer does not incur in economical loss Let:

- μ_C be the price (in monetary units) that the customer pays to the maintainer for each contracted hour of resource;
- μ_{MO} be the price that the maintenance organization pays to the resources. If there were no more economical parameters in the project, then it is obvious that μ_{MO} should be greater than μ_C if the maintainer wants to earn money;
- s be the magnitude of the sanction that the maintenance organization must pay to the customer for each hour of delay in error correction.

If the maintainer dedicates less resources than needed, then it will undoubtedly incur in delays, with the respective sanctions. However, the maintainer will be interested in incurring in such delays if sanctions plus costs of devoting less resources than needed is less than the cost of devoting all the needed resources. Let us study how many resources the maintainer must dedicate and not suffer monetary loss.

Table 6 shows a generic situation of a maintenance project for a period of n days. Except the last row, which shows some totals, the first column represents the day under consideration; the second one is the number of hours that the maintenance organization will dedicate to correct errors (note that it plans p hours for every day). The third column (h_i) is the predicted number of hours that will be needed to fix errors. Supposing that $p \leqslant h_i$, the fourth column shows the number of days needed that the maintenance organization shall have to dedicate to fix the errors occurred in the ith day, supposing that in the following days h_i hours will also be dedicated. The fifth column shows the day when the errors produced in the respective day will be corrected (for example, if $h_i = 8\ \forall i$ and $p = 4$, errors of the first day will be corrected the second day, which means to begin with the delay of the correction of errors of the second day, and so on).

Table 6
Parameters of a generic maintenance project.

Day	Planned hours	Needed hours	Needed days	Day of end	Delay	Sanction
1	p	h_1	$\frac{h_1}{p}$	$\frac{h_1}{p}$	$\frac{h_1}{p} - 1 = \frac{h_1 - p}{p}$	$s \cdot \left(\frac{h_1}{p} - 1\right) = s \cdot \frac{h_1 - p}{p}$
2	p	h_2	$\frac{h_2}{p}$	$\frac{h_1}{p} + \frac{h_2}{p}$	$\frac{h_1 + h_2 - 2p}{p}$	$s \cdot \left(\frac{h_1 + h_2}{p} - 2\right) = s \cdot \frac{h_1 + h_2 - 2p}{p}$
3	p	h_3	$\frac{h_3}{p}$	$\frac{h_1 + h_2 + h_3}{p}$	$\frac{h_1 + h_2 + h_3 - 3p}{p}$	$s \cdot \frac{h_1 + h_2 + h_3 - 3p}{p}$
					...	
n	p	h_n	$\frac{h_n}{p}$	$\sum_{i=1}^{n} \frac{h_i}{p}$	$\frac{\sum_{i=1}^{n} h_i - np}{p}$	$s \cdot \frac{\sum_{i=1}^{n} h_i - np}{p}$
	pn	$\sum_{i=1}^{n} h_i$			$\sum_{j=1}^{n} \frac{\sum_{i=1}^{j} (h_i - ip)}{p}$	$s \cdot \sum_{j=1}^{n} \frac{\sum_{i=1}^{j} (h_i - ip)}{p}$

The delay in the correction of the errors of each day is shown in the sixth column, and its respective sanction (function S) appears in the seventh and last column.

In the maintenance project of table 6, the maintainer will pay $\mu_{\mathrm{MO}}p$ monetary units daily to the contracted resources, whereas the customer will pay $\mu_{\mathrm{C}}h_i$ daily. During the n days, these quantities will be $C(p) = \mu_{\mathrm{MO}}np$ and $I(p) = \sum_{i=1}^{n} \mu_{\mathrm{C}}h_i$, respectively. In order to simplify the problem, let us suppose that h are the daily hours of resource needed (this may introduce small errors in the model if the differences between the h_i are quite large; nevertheless, the length of the planned periods are usually short enough so that there are not these large differences between them. With this consideration, the maintainer gets $I(p) = \mu_{\mathrm{C}}nh$ monetary units from the customer for all the days. Also we can rewrite the first n terms of the last column in this way:

$$s \cdot \frac{h-p}{p}; \quad s \cdot \frac{2\cdot(h-p)}{p}; \quad s \cdot \frac{3\cdot(h-p)}{p}; \quad \ldots; \quad s \cdot \frac{n\cdot(h-p)}{p}.$$

The last cell of the last column (total sanction, dependent on p, which is the daily quantity of resources to allocate by the maintenance organization) is the sum of these terms:

$$\begin{aligned} S(p) &= s \cdot \frac{h-p}{p} + s \cdot \frac{2\cdot(h-p)}{p} + s \cdot \frac{3\cdot(h-p)}{p} + \cdots + s \cdot \frac{n\cdot(h-p)}{p} \\ &= s \cdot \frac{h-p}{p} \cdot (1+2+3+\cdots+n) = s \cdot \frac{h-p}{p} \cdot \sum_{i=1}^{n} i \\ &= s \cdot \frac{h-p}{p} \cdot \frac{n\cdot(n+1)}{2} = \frac{n\cdot(n+1)\cdot s\cdot(h-p)}{2p}. \end{aligned}$$

Now we have all the needed information to calculate p: the maintainer will use p hours of resource ($p \leqslant h$) from the value which evens $I(p)$ with $C(p) + S(p)$. Let us solve this equation:

$$\begin{aligned} &I(p) = C(p) + S(p), \quad p \leqslant h, \\ &n\mu_{\mathrm{C}}h = n\mu_{\mathrm{MO}}p + \frac{n\cdot(n+1)\cdot s\cdot(h-p)}{2p}, \\ &2pn\mu_{\mathrm{C}}h = 2pn\mu_{\mathrm{MO}}p + n\cdot(n+1)\cdot s\cdot(h-p), \\ &2pn\mu_{\mathrm{C}}h = 2pn\mu_{\mathrm{MO}}p + n\cdot(n+1)\cdot s\cdot h - n\cdot(n+1)\cdot sp, \\ &2p^2n\mu_{\mathrm{MO}} - 2pn\mu_{\mathrm{C}}h + n\cdot(n+1)\cdot s\cdot h - n\cdot(n+1)\cdot sp = 0, \\ &2p^2\mu_{\mathrm{MO}} - 2p\mu_{\mathrm{C}}h + (n+1)\cdot s\cdot h - (n+1)\cdot sp = 0, \\ &2p^2\mu_{\mathrm{MO}} - p\bigl[2\mu_{\mathrm{C}}h + (n+1)\cdot s\bigr] + (n+1)\cdot s\cdot h = 0, \quad p \leqslant h. \end{aligned} \tag{1}$$

If the maintainer solves this equation with the parameters of its maintenance project, then it will know the quantity of resources to be dedicated to not incur in eco-

Table 7
Function values according to p.

p	$I(p)$	$C(p)$	$S(p)$	$Bmo(p)$
1	36000	1500	1302000	−1267500
2	36000	3000	558000	−525000
3	36000	4500	310000	−278500
4	36000	6000	186000	−156000
5	36000	7500	111600	−83100
6	36000	9000	62000	−35000
7	36000	10500	26571	−1071
7.02	*36000*	*10530*	*25966*	*−496*
7.04	*36000*	*10560*	*25364*	*76*
8	36000	12000	0	24000
9	36000	13500	−20667	43167

nomical loss (p). The maintainer obtains as profit the difference between the income and the sum of costs plus sanctions; this is:

$$Bmo(p) = I(p) - C(p) - S(p).$$

In table 7 we have represented the economical functions for a maintenance project with the following parameters:

- $\mu_{MO} = 50$ monetary units per hour really dedicated, paid by the maintainer to resources;
- $\mu_C = 150$ monetary units per contracted hour, paid by the customer to the maintainer;
- $n = 30$ days, which is the duration of planned period;
- $s = 400$ monetary units per day of delay, paid by the maintainer to the customer;
- $h = 8$ hours per day, which is the quantity of required resources (estimated by the maintainer).

We observe that the optimal quantity of resources to be dedicated is 8 hours/day, which coincides with the number of hours needed. Obviously, to dedicated more than 8 hours per day does not produce, as can be understood from table 7, that sanctions be now profits for the maintainer (since there are no sanctions from this point), and we must reject these values. Solving equation (1) for this case, we obtain the two following values for p:

$$p_1 = 7.037 \text{ hours/day}; \quad p_2 = 140.96 \text{ hours/day}.$$

The second value is rejected since it is greater than h. From p_1, the maintainer knows that it can dedicate during this period approximately one hour less than those required with no economical loss. The remaining hour can be used in other more critical projects in the considered period. This is a good mechanism to deal with resources planning and estimation, as currently, when there is a notorious lack of qualified people in Information Technologies.

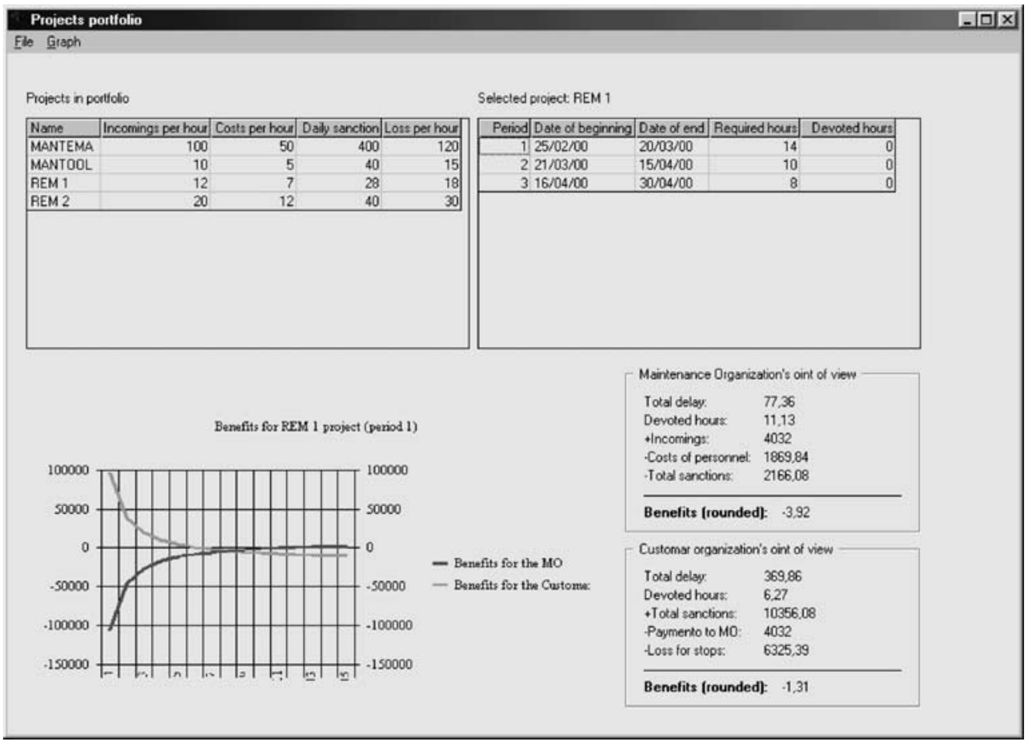

Figure 4. Calculated resources of some projects.

This technique has been implemented in SREM (figure 4), a Single tool for Resources Estimating of Maintenance projects, which maintains the economical status of an applications portfolio.

Data obtained with this technique are also very useful for the negotiation of the contract. They can also be used by the customer, but in this case some parameters will change their meaning (for example, sanctions paid by the maintainer will be now profits for the customer), and a new one must be added to the model (the economical loss per hour of system stop).

4.1.3. Task 0.3: Contract

Once the maintenance proposal has been analyzed and discussed by both parts, a contract detailing at least the terms indicated in figure 5 is signed:

4.2. Activity 1: Process implementation

Although in this task the customer and the maintenance organizations have already signed the maintenance service contract, the maintainer is not yet responsible for the execution of the maintenance interventions. Such responsibility will arrive at the end of this activity, once its tasks have been fully executed. The following are the tasks in this activity:

1. Identification of the parts
2. Object of the contract (*the customer entrusts the supplier a set of features in the field of the computer services that has as purpose the maintenance of the software registered in the context of the technical proposal*)
3. Characteristics of the presentation of the service
   3.1. Inventory of the software objects to maintain
   3.2. Initial State of the software
   3.3. Conditions of organization of the work of the supplier and of the customer
   3.4. Formalization conditions of the maintenance intervention
   3.5. Adaptation of the quality assurance plan to the customer
   3.6. Correction of anomalies
   3.7. Maintenance of the competitive level of the application software and of the standard software
   3.8. Documentation facilitated to the attached presentation
   3.9. Assistance modality to the final users
4. Obligations of the customer
5. Obligations of the supplier
6. Clauses of exclusion
7. Clauses of organization
8. Guarantee of the interventions
   8.1. Data recovery
   8.2. Responsibilities definition
9. Other clauses

Figure 5. Contents of the maintenance contract.

4.2.1. Task 1.0: Planning customer/supplier relations

In this task, a calendar detailing the future control-meetings must be set up. The purpose of these meetings is to let the customer know the status of its applications portfolio, although they also can be understood as milestones for controlling the maintenance organization's work.

Besides, people responsible for each organization (customer, user and maintainer) must be identified, determining who will represent each profile mentioned in table 2.

4.2.2. Task 1.1: Knowledge acquisition

In this task, the maintenance team (a maintainer's profile) studies the existing documentation on the software to be maintained, including source code, database schemes, etc. They must also hold interviews with users and pay attention to how the current maintenance organization works. During this task, the customer (or its cur-

1. Introduction
   1.1. Scope and purpose of the document
   1.2. Project goals
      1.2.1. Goals
      1.2.2. Main functions
      1.2.3. Technical and management constraints
2. Project estimations
   2.1. Summary of the data obtained in questionnaires (number of programs, size, use of standards, etc.)
   2.2. Historic data (if there are)
   2.3. Estimation techniques and metrics
   2.4. Metric values
3. Project resources
   3.1. Human resources
   3.2. Hardware and software
   3.3. Other resources
4. Personnel organization
5. Tracking and monitoring mechanisms
6. Appendix (if there are)

Figure 6. Contents of the Technical Summary.

rent maintenance organization) must resolve maintenance requests, since the new maintenance organization is now learning the software's structure and operation mode.

This task takes approximately between one and two months, and during this period the new maintainer will probably not have modified one line of code. At the end of this task, the new maintainer must provide the customer updated documentation on the software.

On the other hand, this task may be partially omitted if the software has not been maintained previously.

4.2.3. Task 1.2: Developing plans

The maintenance team develops the maintenance plans and builds-up the Technical Summary (figure 6) of the software, once they have acquired the knowledge. The Technical Summary must collect information on project goals, main functions, technical and management constraints, a summary of the data obtained in the questionnaires, historical data, estimating techniques and metrics, metric values, control mechanisms, etc.

4.2.4. Task 1.3: Defining modification requests procedures
The maintenance organization generates document templates for modification requests (MANTEMA proposes a generic one) and, with the customer, defines the procedures for their presentation.

Procedures agreed on in this task will be used in the future delivery of modification requests.

4.2.5. Task 1.4: Implementing Software Configuration Management process
As we can see in table 1, an interface with the Configuration Management process is established in this task. The CM process may be tailored from the customer's if it already has one, or from the standard CM process of maintainer if otherwise.

4.2.6. Task 1.4: Preparing tests environments
The maintenance team must prepare copies of the software product for the future software intervention tests.

From this moment on, the new maintenance organization will be responsible for the execution of all the maintenance interventions, from its reception until its integration in the production environment.

4.3. Middle activities and tasks of the MANTEMA maintenance process

Now, all maintenance requests arrive directly to the Maintenance Request Manager (MRM), which is the profile in charge of the reception of maintenance requests. This profile classifies and directs the modification request to one of the contracted maintenance types (either rejects or reconsiders it with the customer if it does not belong to any one of them). After this, the Scheduler (another profile, but often also assigned to the MRM) puts it into the modification request queue associated to the corresponding Maintenance Team (another profile of the maintainer), adequately placing it according to its priority and importance.

When the Maintenance Team receives the modification request, it enters the set of activities and tasks of its maintenance type [23].

4.4. Common final activities and tasks

Four activities compose this last node of the graph shown in figure 1, although only the last one (*End of outsourcing*) is of our interest, from the outsourcing point of view (the first three are *Intervention register*, *Historical database updating* and *Retirement*).

This last activity is executed at the end of the outsourcing period and it comprises the following three tasks.

4.4.1. Task F5.1: Inventory and documentation delivery
Depending on the contract, the maintenance organization might have to deliver to the customer all the software products generated and modified during the period in which it has been responsible for the maintenance process.

4.4.2. Task F5.2: Training and return of experience
This task is the inverse of task 1.1 (*Knowledge acquisition*), during which the Maintenance Team learned the characteristics of the system by watching the work mode of the maintainer that, at that moment the customer had. Now, it is the maintainer who must train the new maintenance staff, allowing maintenance with mixed teams.

4.4.3. Task F5.3: Final delivery of the service
The maintenance organization definitively finishes the services rendered to the customer.

5. Conclusions and future lines of work

In this paper we have presented the outsourcing strategy contemplated in MANTEMA, a methodology for software maintenance. MANTEMA has been jointly developed with Atos ODS, the third largest European organization in software services, and specialized in outsourcing of software maintenance.

An important contribution of MANTEMA is the integration of outsourcing activities in the maintenance process. According to [20], process models based on products are not applicable to processes based on services. In the context of outsourcing, it is obvious that maintenance is understood as a service contracted between two organizations.

Moreover, MANTEMA has been a cornerstone for our partner's recent ISO 9000 certification. But the possession of a standard, measurable software process is also a requisite condition to get meaningful levels of maturity in others models, as CMM or SPICE. MANTEMA provides this with a standard process model and with a wide set of metrics (not only Service Level Agreements). In fact, the use of MANTEMA as a baseline for evaluating maintenance processes has been recently analyzed in [27].

Software process audits are also important (and most specially in the case of outsourcing), as is evidenced by the inclusion of this process in the ISO/IEC 12207 Standard: in [31] the integration of the audit process into the maintenance and into MANTEMA is studied.

Acknowledgements

The authors wish to thank the anonymous reviewers whose suggestions have considerably improved the paper.

This work is 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] A.J. Albretch, Measuring application development productivity, in: *Proceedings of the IBM Application Development Symposium*, Monterrey, Canada (1977) pp. 83–92.

[2] V. Basili, L. Briand, S. Condon, Y. Kim, W. Melo and J.D. Valett, Understanding and predicting the process of software maintenance releases, in: *Proceedings of the International Conference on Software Engineering* (IEEE Computer Society, Los Alamitos, CA, 1996) pp. 464–474.

[3] L.C. Briand, K. El Emam and F. Bomarius, COBRA: a hybrid method for software cost estimation, benchmarking and risk assessment, in: *Proceedings of the 20th International Conference on Software Engineering* (IEEE Computer Society Press, Los Alamitos, CA, 1998) pp. 390–399.

[4] J.M. Brower, Outsourcing and privatizing information technology, Crosstalk, The Journal of Defense Software Engineering (1997) 28–30.

[5] F. Calzolari, P. Tonella and G. Antoniol, Modelling maintenance effort by means of dynamic systems, in: *Proceedings of the 3rd European Conference on Software Maintenance and Reengineering*, Amsterdam (The Netherlands) (IEEE Computer Society, Los Alamitos, CA, 1998) pp. 150–156.

[6] D.N. Card and R.L. Glass, *Measuring Software Design Quality* (Englewood Cliffs, 1990).

[7] L. De Looff, *Information Systems Outsourcing Decision Making: A Managerial Approach* (Idea Group Publishing, Hershey, PA, 1990).

[8] M. De Vogel, Outsourcing and metrics, in: *Proceedings of the 2nd European Measurement Conference, FESMA'99* (Federation of European Software Metrics Association/Technologisch Instituut, 1999) pp. 217–225.

[9] M.J. Earl, *Management Strategies for Information Technology* (Prentice-Hall, London, UK, 1989).

[10] Euromethod Project, Euromethod version 1 (1996).

[11] I. Graham, B. Henderson-Sellers and H. Younessi, *The OPEN Process Specification* (ACM Press and Addison-Wesley, Essex, UK, 1997).

[12] T. Hoffman, Users say move quickly when outsourcing your personnel, Computer World (March 1997) 77.

[13] IEEE Std. 1219-1992, *IEEE Standard for Software Maintenance* (The Institute of Electrical and Electronics Engineers, Inc. New York, USA, 1992).

[14] ISO/IEC 12207 – International Organization for Standardization and International Electrotechnical Commission, *Information Technology. Software Life Cycle Processes*, Canada (1995).

[15] D.W. Karolak, *Software Engineering Risk Management* (IEEE Computer Society Press, Los Alamitos, CA, 1996).

[16] R. Klepper and W.O. Jones, *Outsourcing Information Technology, Systems and Services* (Prentice-Hall, New Jersey, 1998).

[17] H.-J. Kung and Ch. Hsu, Software maintenance life cycle model, in: *Proceedings of the International Conference on Software Maintenance* (IEEE Computer Society, Los Alamitos, CA, 1998) pp. 113–121.

[18] A.C. Neitzel, Managing risk management, Crosstalk, The Journal of Defense Software Engineering (July 1999) 17–21.

[19] F. Niessink and H. van Vliet, Predicting maintenance effort with function points, in: *Proc. of the International Conference on Software Maintenance* (IEEE Computer Society, Los Alamitos, CA, 1997) pp. 32–39.

[20] F. Niessink and H. van Vliet, The IT service capability maturity model, Technical Report IR-463, Release L2-1.0. Available on-line (2000-10-31), http://www.cs.vu.nl/~frankn/publications/TR99.Niessink.pdf (1999).

[21] M. Paulk, C. Weber and B. Curtis, *The Capability Maturity Model: Guidelines for Improving the Software Process* (Addison-Wesley, Reading, MA, 1995).

[22] T.M. Pigoski, *Practical Software Maintenance. Best Practices for Managing Your Investment* (Wiley, USA, 1997).

[23] M. Polo, M. Piattini, F. Ruiz and C. Calero, MANTEMA: A complete rigorous methodology for supporting maintenance based on the ISO/IEC 12207 Standard, in: *Proc. of the 3rd European Conference on Software Maintenance and Reengineering*, eds. Nesi and Verhoef (IEEE Computer Society, Los Alamitos, CA, 1999) pp. 178–181.

[24] M. Polo, M. Piattini, F. Ruiz and C. Calero, Using the ISO/IEC 12207 tailoring process for defining a maintenance methodology, in: *Proc. of the 1st IEEE Conference on Standardisation and Innovation in Information Technology*, Aachen, Germany, eds. Jakobs and Williams (IEEE Computer Society, Los Alamitos, CA, 1999) pp. 205–210.

[25] M. Polo, M. Piattini, F. Ruiz and C. Calero, Roles in the maintenance process, Software Engineering Notes 24(4) (1999) 84–86.

[26] M. Polo, M. Piattini and F. Ruiz, *Managing the Maintenance Process, World Class IT Service Management Guide*, ed. Jan van Bon (ten Hagem & Stam Publishers, The Hague, The Netherlands, 2000).

[27] M. Polo, M. Piattini and F. Ruiz, Improving the quality of the maintenance process, in: *Proc. of the 2nd World Congress on Software Quality*, Tokyo, Japan (2000) pp. 325–330.

[28] R.S. Pressman, *Software Engineering: A Practitioner's Approach* (McGraw-Hill, 1993).

[29] Project Management Institute, *A Guide to the Project Management Body of Knowledge* (Project Management Institute, 1996).

[30] H.R. Rao, K. Nam and A. Chaudhury, Information systems outsourcing, Communications of the ACM 39(7) (1996) 27–28.

[31] F. Ruiz, M. Piattini, M. Polo and C. Calero, Audit of software maintenance process, in: *Auditing Information Systems*, ed. Piattini (Idea Group Publishing, Hershey, PA, 2000).

[32] N.F. Schneidewind, Measuring and evaluating maintenance process using reliability, risk and test metrics, in: *Proceedings of the 19th International Conference on Software Maintenance* (IEEE Computer Society Press, Los Alamitos, CA, 1997) pp. 232–239.

[33] N.F. Schneidewind, How to evaluate legacy systems maintenance, IEEE Software (July/August 1998) 34–42.

[34] M. Shepper, C. Schofield and B. Kitchenham, Effort estimation using analogy, in: *Proceedings of the 18th International Conference on Software Maintenance* (IEEE Computer Society Press, Los Alamitos, CA, 1996) pp. 170–178.

[35] S.A. Sherer, Using risk analysis to manage software maintenance, Journal on Software Maintenance: Research and Practice 9 (1997) 345–364.

[36] J. Van Bon, Sourcing, in: *World Class IT Service Management Guide*, ed. Jan van Bon (ten Hagem & Stam Publishers, The Hague, The Netherlands, 2000).