Available online at www.sciencedirect.com
SCIENCE DIRECT®
The Journal of Systems and Software 77 (2005) 113–129
The Journal of Systems and Software
www.elsevier.com/locate/jss

A family of experiments to validate metrics for software process models

G. Canfora [a], F. García [b,*], M. Piattini [b], F. Ruiz [b], C.A. Visaggio [a]

[a] RCOST – Research Centre on Software Technology, Dipartimento di Ingegneria, Università del Sannio, Palazzo ex Poste, viale Traiano, 82100 Benevento, Italia

[b] Alarcos Research Group, University of Castilla-La Mancha, Paseo de la Universidad, 4, 13071, Ciudad Real, Spain

Received 5 July 2004; received in revised form 2 November 2004; accepted 14 November 2004
Available online 18 December 2004

Abstract

Process modelling is a key activity of software process management and it is the starting point for enacting, evaluating and improving software processes. The current competitive marketplace calls for the continuous improvement of processes and therefore, it is fundamental to have software process models with a high maintainability. In this paper we introduce a set of metrics for software process models and discuss how these can be used as maintainability indicators. In particular, we report the results of a family of experiments that assess relationships between the structural properties, as measured by the defined metrics, of the process models and their maintainability.

Keywords: Software process models maintainability; Metrics; Experimental software engineering

1. Introduction

The management of software processes is largely recognized as a key factor for improving both the productivity of an organization and the quality of the software delivered (Fuggetta, 2000). Process Modelling is an important activity of software process management and it is the starting point for analyzing, improving, and enacting processes (Florac and Carleton, 1999).

A software process model (SPM) is an abstraction of a real-world software process expressed in a suitable process modelling language (PML). SPMs applications range from comprehension to enactment; Curtis et al. (1992) identify five main applications of process modelling:

- To facilitate human understanding and communication.
- To support process improvement.
- To support process management.
- To automate guidance in performing process.
- To automate execution support.

SPMs can be grouped into two main categories: descriptive models and active models (Curtis et al., 1992; Dowson and Fernström, 1994). Descriptive models are aimed at describing processes and organizational behaviour in terms of entities—activities, roles, tools, and artifacts- and the relationships among them. Active models are intended for building executable systems that support the enactment of processes.

Descriptive SPMs, which are the focus of this paper, have proven useful for guiding process execution and as a basis for measurement in the context of software process improvement (Becker-Kornstaedt, 2000). They are a prerequisite for active modelling.

* Corresponding author. Tel.: +34 926295300/3708; fax: +34 926295354.
E-mail address: felix.garcia@uclm.es (F. García).

0164-1212/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2004.11.007

The current competitive marketplace forces software organizations to continuously improve their processes. Improving software processes effectively requires their maintenance: maintenance of the software process deserves the same attention as any other kind of software (Curtis, 1992). As a consequence, process models must be continuously maintained based on gained experience, new requirements and changed policies (Jaccheri and Conradi, 1993). This suggests the need for descriptive software process models with high maintainability. [1] In particular, means are needed to evaluate the maintainability of software processes in the early stages of their development, primarily during process modelling. This would provide organizations with a basis for choosing, among semantically equivalent SPMs, the model which can be more easily maintained and adapted to new and emerging needs.

In this context, this paper addresses two main issues:

- To quantify descriptive SPMs by means of the definition of a set of suitable metrics.
- To demonstrate which metrics can be used as maintainability indicators by carrying out an empirical study.

This paper is organized as follows. Section 2 provides an overview of related work and Section 3 introduces the metrics for SPMs and presents an example of computation. Section 4 provides an overview of the family of experiments carried out in order to empirically validate the metrics. Sections 5 and 6 describe the individual experiments and Section 7 presents a global analysis of the results. Finally, conclusions and future works are outlined in Section 8.

2. Related works

Software process research has gained a great importance due to the growing interest of software companies in the improvement of the productivity and quality of delivered products. To support software process evaluation and improvement, a wide variety of initiatives have proposed reference frameworks. Notable examples are the CMM (SEI, 1995), the CMMI (SEI, 2002), the ISO 15504 standard (ISO/IEC, 1998). Process improvement has also been considered in the new family of ISO 9000:2000 standards (ISO/IEC, 2000a,b). In all these initiatives measurement plays a fundamental role as a means for assessing and institutionalising software process improvement programs. Basically, three types of entities can be measured: SPMs, projects, and products.

Research on software process measurement has focused mainly on the measurement of projects—in terms of cost and schedule- and products. For SPMs, an important quality criteria evaluated is accuracy, intended as the degree to which the model reflects the actual process (Cook and Wolf, 1999).

In this paper we focus on the evaluation of the maintainability of SPMs.

Software maintainability has been broadly treated in literature and there have been several efforts in recent years to characterize and quantify it. A wide variety of works have studied software product maintainability as a dependent variable and most of them have focused on code. In the context of object-oriented systems, Li and Henry (1993), Harrison et al. (2000), and Briand et al. (2001) have developed prediction models for maintenance tasks. Several authors have investigated the evaluation of maintainability in the early stages of the development of the software product. Some relevant works related with the study of the maintainability of UML models are:

- Marchesi (2003) and Saeki (2003), which proposes metrics for use case diagrams;
- Genero et al. (2003), which introduces metrics for class diagrams;
- Miranda et al. (2003), which defines metrics for state-chart diagrams.

Maintainability of SPMs is a relevant issue to consider with the aim of finding useful early process maintainability indicators. It may provide the quantitative basis for easing the changes and evolution of the models in the context of process improvement and constitutes the main motivation and goal of this current work.

3. Metrics for software process models

A representative set of metrics for software process models has been defined in order to evaluate SPMs maintainability (Table 1).

The metrics have been defined following the SPEM (Software Process Engineering Metamodel) terminology (OMG, 2002), but they can be easily applied to other PMLs. Fig. 1 shows an example of a software process model represented with the SPEM; in particular, it consists of a UML Activity Diagram with stereotypes which represent the SPEM constructors.

Table 2 shows the values of the metrics for the example illustrated in Fig. 1.

The metrics defined are model scope metrics, as they measure the structural properties of the overall software process model. If we take into account that a software process model with a high degree of structural complex-

[1] In analogy to the definition of software maintainability (IEEE, 1990), we intend the maintainability of a software process as the ease with which it can be understood, corrected, adapted, and enhanced.

Table 1
Model scope metrics

Metric	Definition
NA	Number of *activities* of the software process model
NWP	Number of *work products* of the software process model
NPR	Number of *roles* which participate in the process
NDWPIn	Number of input dependences of the *work products* with the *activities* in the process
NDWPOut	Number of output dependences of the *work products* with the *activities* in the process
NDWP	Number of dependences between *work products* and *activities* $\mathrm{NDWP(PM)} = \mathrm{NDWPIn(MP)} + \mathrm{NDWPOut(MP)}$
NDA	Number of precedence dependences between *activities*
NCA	Activity coupling in the process model. $\mathrm{NCA(PM)} = \frac{\mathrm{NA(PM)}}{\mathrm{NDA(PM)}}$
RDWPIn	Ratio between *input dependences* of work products with activities and *total number of dependences* of work products with activities $\mathrm{RDWPIn(PM)} = \frac{\mathrm{NDWPIn(PM)}}{\mathrm{NDWP(PM)}}$
RDWPOut	Ratio between *output dependences* of work products with activities and *total number of dependences* of work products with activities $\mathrm{RDWPOut(PM)} = \frac{\mathrm{NDWPOut(PM)}}{\mathrm{NDWP(PM)}}$
RWPA	Ratio of *work products and activities*. Average of the work products and the activities of the process model. $\mathrm{RWPA(PM)} = \frac{\mathrm{NWP(PM)}}{\mathrm{NA(PM)}}$
RRPA	Ratio of *process roles* and *activities* $\mathrm{RRPA(PM)} = \frac{\mathrm{NPR(PM)}}{\mathrm{NA(PM)}}$

ity is much more difficult to maintain, hence these metrics can be considered a good maintainability indicator. This hypothesis transposes the relationship between structural complexity and maintainability of software artefacts (Briand et al., 1998) to the domain of software processes, as depicted in Fig. 2.

As we can observe in Fig. 2, the metrics evaluate different structural properties of a SPM, namely size, complexity, and coupling, according to the theoretical validation results obtained by applying the Briand et al. (1996) framework. The figure shows that the structural properties of the SPMs affect their maintainability. Accordingly, our aim is to understand which metrics can be used as SPMs maintainability indicators. We have focused on three relevant sub-characteristics of maintainability:

Table 2
Values of model level metrics

Metric	Value	Metric	Value
NA	6	NDA	6
NWP	6	NCA	$6/6 = 1$
NPR	3	RDWPIn	$5/11 = 0.455$
NDWPIn	5	RDWPOut	$6/11 = 0.545$
NDWPOut	6	RWPA	$6/6 = 1$
NDWP	11	RRPA	$3/6 = 0.5$

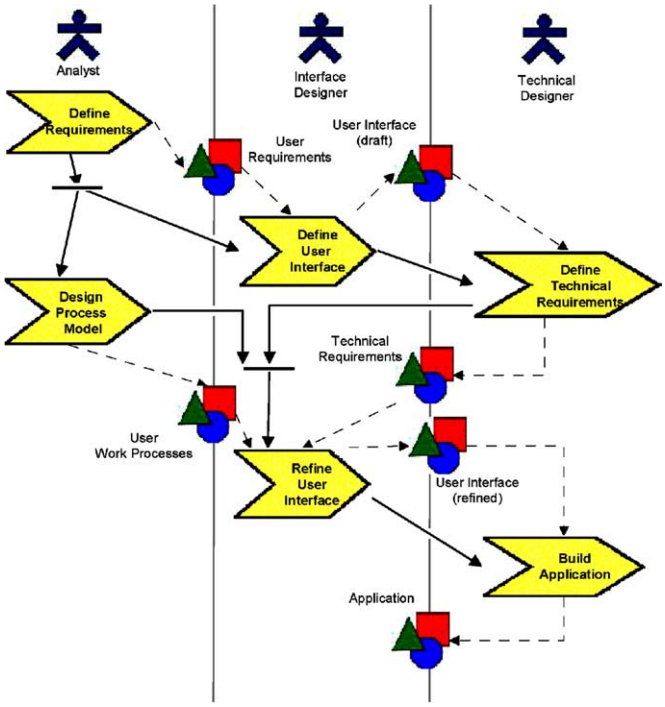

Fig. 1. Example of a software process model represented with the SPEM.

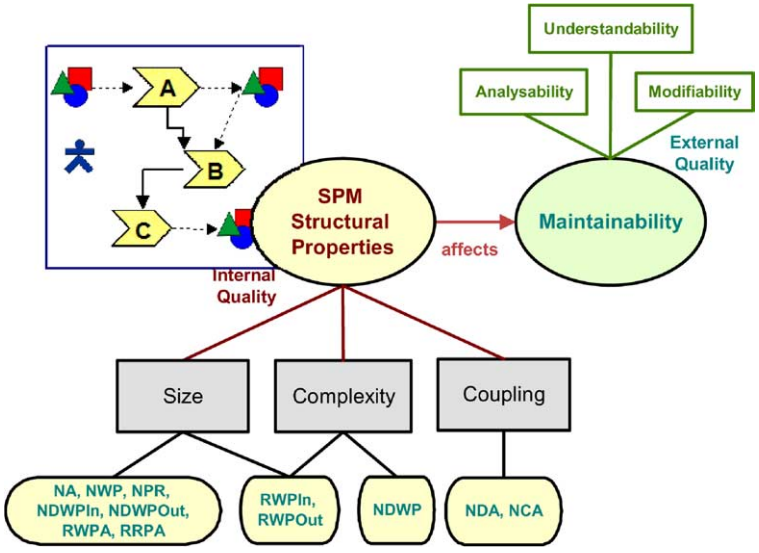

Fig. 2. Relationship between structural complexity and maintainability.

- *Analyzability*. Easiness shown by the model in discovering errors or deficiencies and in guessing the parts that should be modified.
- *Understandability*. Easiness with which the model can be understood.
- *Modifiability*. Easiness with which the model can be modified, for possible errors, a specific modification request or new requirements.

In order to understand which metrics can be successfully used to evaluate maintainability, a family of experiments was carried out which is described in the following sections.

4. Overview of the family of experiments

A family of experiments contains multiple similar empirical studies pursuing the same goal. As Basili et al. (1999) remark, a family of experiments permits the accumulation of the knowledge needed to extract significant conclusions that can be applied in practice. We planned and performed a family of experiments based on the methodology of Ciolkowski et al. (2002). A five steps process was exploited:

1. *Experiment preparation*. The general goal of the experiments was to demonstrate the suitability of the selected SPM metrics as maintainability indicators. By using the GQM template (Basili and Rombach, 1988) the experiment goal can be defined as follows:
   - Analyze SPMs Metrics.
   - With the purpose of Evaluating.
   - With respect to their capability of being used as maintainability indicators.
   - From the point of view of the Researchers.
   - In the context of Computer Science Undergraduate Students and Professionals of Information Systems.
2. *Context definition*. In order to ease the generalization of the results the following groups of subjects were identified to establish the context of each individual experiment:
   - Professionals. They are the ideal subjects to generalize the results, and for this reason we have to use this kind of subjects whenever it is possible.
   - Students. They play a very important role in software engineering experimentation, because in general before performing studies in industrial environments, which requires resources, and time, researchers carry out pilot studies with students in academic environments (Carver et al., 2003). In addition, students are the next generation of professionals (Kitchenham et al., 2002) and under some conditions, there is not a great difference between students and professionals. In situations in which the tasks to perform do not require industrial experience, experimentation with students is viable (Höst et al., 2000; Basili et al., 1999).
3. *Material*. The material prepared for the family of experiments was composed of eighteen SPMs with different metric profiles, as shown in Table 3. The models are based on different methodologies and SPMs found in literature, as for example PMBOK, Rational Unified Process, etc. Additional material was prepared for the individual experiments based on the types of tasks to be performed on the models and the data to be gathered.
4. *Conduct individual experiments*. According to the general plan of the family we carried out five individual experiments, as shown in Fig. 3.

Table 3
Values of model level metrics for the 18 SPMs which constitute the material of the family of experiments

Mod	NA	NWP	NPR	NDWPIn	NDWPOut	NDWP	NDA	NCA	RDWPIn	RDWPOut	RWPA	RRPA
1	6	6	3	5	6	11	6	1.000	0.455	0.545	1.000	0.500
2	5	6	4	5	5	10	4	1.250	0.500	0.500	1.200	0.800
3	2	13	2	12	3	15	1	2.000	0.800	0.200	6.500	1.000
4	9	25	9	25	21	46	11	0.818	0.543	0.457	2.778	1.000
5	5	6	4	5	5	10	8	0.625	0.500	0.500	1.200	0.800
6	4	11	4	14	9	23	3	1.333	0.609	0.391	2.750	1.000
7	8	17	1	15	11	26	9	0.889	0.577	0.423	2.125	0.125
8	5	8	4	13	5	18	4	1.250	0.722	0.278	1.600	0.800
9	7	12	1	12	11	23	6	1.167	0.522	0.478	1.714	0.143
10	24	37	10	72	40	112	24	1.000	0.643	0.357	1.542	0.417
11	7	12	5	12	11	23	6	1.167	0.522	0.478	1.714	0.714
12	2	8	3	6	4	10	1	2.000	0.600	0.400	4.000	1.500
13	3	6	1	8	3	11	4	0.750	0.727	0.273	2.000	0.333
14	3	5	7	5	3	8	2	1.500	0.625	0.375	1.667	2.333
15	4	9	1	9	7	16	6	0.667	0.563	0.438	2.250	0.250
16	8	6	4	9	9	18	7	1.143	0.500	0.500	0.750	0.500
17	4	24	1	20	11	31	3	1.333	0.645	0.355	6.000	0.250
18	5	21	3	21	11	32	4	1.250	0.656	0.344	4.200	0.600

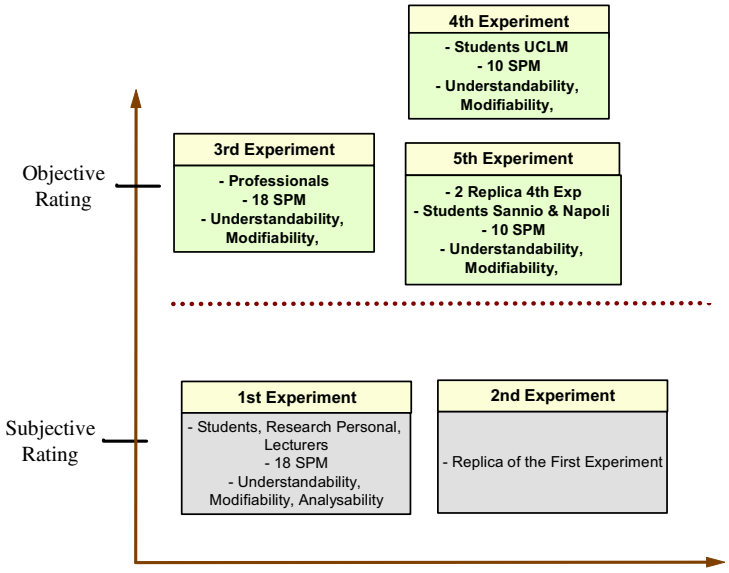

Fig. 3. Experiments of the family.

The individual experiments were grouped under two main categories depending on the kind of tasks to be performed by the subjects:

- *Subjective rating*. In this group, the maintainability sub-characteristics are rated in a subjective way according to the opinion of the subjects.
- *Objective rating*. In the objective experiments the subjects have to perform a set of tasks on the models related to their maintainability (understandability and modifiability). In these experiments the dependent variables are measured in an objective way by calculating the time spent by the subjects in performing these tasks.

To carry out each individual experiment we took into account the general plan established in the context of the experiment family and the feedback obtained as a result of performing each experiment of the family. These experiments are described with more detail in Sections 5 and 6.

5. *Family data analysis*. When the result data of the individual experiments is collected and analyzed, not only it is important to obtain local conclusions (related to individual experiments), but, it is fundamental to extract the overall conclusions obtained with the carrying out of the experiment family. This analysis is described in Section 8.

5. Subjective experiments

In this section we describe the subjective experiments. We followed the suggestions provided by Wohlin et al. (2000), Kitchenham et al. (2002), Perry et al. (2000) and Briand et al. (1999) on how to perform controlled experiments. To describe the experiment we use (with only minor changes) the format proposed by Wohlin et al. (2000) comprising the following main tasks: definition, planning, operation, analysis and interpretation, validity evaluation and presentation and package.

5.1. First experiment

5.1.1. Definition

The goal of the experiment is defined as: *Analyze* metrics of the SPMs structural properties *for the purpose of* evaluating *with respect to* their capability of being used as SPM maintainability indicators *from the point of view of* the researchers *in the context of* Undergraduate Computer Science students and professors of the Software Engineering area at the Department of Computer Science at the University of Castilla-La Mancha.

5.1.2. Planning

Context selection. The context of the experiment was a group of 20 subjects (students, researchers and assistant professors) belonging to the Software Engineering Area at the Department of Computer Science at the University of Castilla-La Mancha in Spain. The experiment was run off-line (not in an industrial software development environment). The experiment was specific since it focused on metrics for SPM structural properties. The experiment addressed a real problem, i.e., which indicators can be used for the maintainability of SPMs? With this purpose we investigated the correlation between SPMs metrics and maintainability sub-characteristics.

Selection of subjects. The subjects were chosen for convenience. The subjects were undergraduate students, researchers and assistant professors who had wide experience and knowledge in software product modelling (UML, databases, etc.), but they did not have any previous experience or knowledge in the modelling of SPMs. A training session was developed to provide the subjects with the necessary knowledge to do the tasks required in the experiment.

Variable selection. The independent variables are the structural properties of SPMs. The dependent variables are three maintainability sub-characteristics: *understandability*, *analyzability*, and *modifiability*.

Instrumentation. The material provided to the subjects consisted of a guide explaining SPEM and 18 SPMs. The independent variables were measured through the metrics proposed at process model scope (see Table 1). The maintainability sub-characteristics were measured according to the subject's ratings. Understandability was also measured using time values. We called this time *understanding time*, that is the time needed to understand a SPM. To measure this time we asked subjects to write down the exact time (hh:mm:ss) when they started to observe the model and the exact time when they finished the exercise.

Hypothesis formulation. We wished to test the following hypotheses:

- *Null hypothesis*, H_0: There is no significant correlation between the metrics and the subject's rating of the three maintainability sub-characteristics: analyzability, understandability, and modifiability.
- *Alternative hypothesis*, H_1: There is a significant correlation between the metrics and the subject's rating of three maintainability sub-characteristics.
- *Null hypothesis*, H_{0t}: There is no significant correlation between the metrics and the understanding time.
- *Alternative hypothesis*, H_{1t}: There is a significant correlation between the metrics and the understanding time.

Experiment design. We selected a within-subject design experiment, i.e., all the tests (experimental tasks) had to be solved by each of the subjects. The tests were put in a different order for each subject.

5.1.3. Operation

Preparation. Subjects were given an intensive training session before the experiment took place. However, the subjects were not aware of what aspects we intended to study. Neither were they aware of the actual hypothesis stated. Each model handed to the subjects was accompanied by a data collection form (see Appendix A) which included the description of three maintainability sub-characteristics: understandability, analyzability, and modifiability. Each subject had to rate each sub-characteristic using a scale consisting of seven linguistic labels. The labels were established according to the suggestions of Godo et al. (1989) and Bonissone (1982).

Execution. The subjects were given all the materials described in the previous paragraph. We explained to them how to carry out the tests. We allowed subjects one week to do the experiment, i.e., each subject had to carry out the test alone, and could use unlimited time to solve it. We collected all the forms with the subjects' rating and times.

Data validation. We checked the forms for completeness. All the forms were complete.

5.1.4. Analysis and interpretation

The data collected was summarized. We had the metric values computed for each SPM, and we calculated the median of the subjects' rating for each maintainabil-

ity sub-characteristic and the mean of the understanding time for each model. We applied the Kolmogorov–Smirnov test to ascertain if the distribution of the collected data was normal. As the data were non-normal, we decided to use a non-parametric test, namely Spearman's correlation coefficient, with a level of significance $\alpha = 0.05$, which means a 95% level of confidence.

Using Spearman's correlation coefficient, each of the metrics was correlated separately to the median of the subject's rating of understandability, analyzability, and modifiability and with the understanding time (see Table 4).

For a sample size of 18 (median values for each SPM) and $\alpha = 0.05$, the Spearman cutoff for accepting H_0 is 0.4684. Analyzing Table 4 we can conclude that there is a significant correlation (rejecting the null hypothesis) between the three maintainability sub-characteristics of the SPM and the metrics NA, NWP, NDWPIn, NDWPOut, NDWP and NDA because the correlation coefficient is greater than 0.4684. The metric RRPA seems to be less correlated with regard to the prior metrics, although it has a correlation value near to the cut-off. The metrics NPR, NCA, RDWPIn, RDWPOut and RWPA seem not to be correlated with maintainability. With regard to the time spent by subjects in the comprehension of the models we can say that there is a correlation (rejecting H_{0t}) with the metrics NA, NWP, NDWPIn, NDWPOut and NDWP. The metric NDA could also be correlated because the correlation value is near to the cut-off.

5.1.5. Validity evaluation

The empirical study's possible threats to validity and the way we attempted to alleviate them were:

Threats to conclusion validity. The only issue that could affect the statistical validity of this study was the size of the sample data (360 values, 18 models and 20 subjects), which is perhaps not enough for both parametric and non-parametric statistic tests (Briand et al., 1995). We are aware of this, so the results of this experiment were considered as preliminary findings.

Threats to construct validity. The dependent variables were three maintainability sub-characteristics: understandability, analyzability, and modifiability. We proposed subjective metrics for them (using linguistic variables), based on the judgement of the subjects. The construct validity of the metrics used for the independent variables is guaranteed by the DISTANCE framework (Poels and Dedene, 2000) used to theoretically validate them.

Threats to internal validity. The following issues were dealt with:

- *Differences among subjects*. Using a within-subjects design, error variance due to differences among subjects is reduced.
- *Knowledge of the universe of discourse among SPMs*. SPMs were from different universes of discourse but general and well-known enough to be familiar to the subjects. Consequently, knowledge of the domain did not affect the internal validity.
- *Accuracy of subject responses.* Subjects assumed responsibility for rating each maintainability sub-characteristic. As they had wide experience in product modelling by mapping this experience to the process modelling, we think their responses could be considered valid. However, we are aware that not all of them had exactly the same degree of experience.
- *Learning effects*. The subjects were given the test in a different order to cancel out learning effects. Subjects were required to answer in the order in which the tests appeared.
- *Fatigue effects*. On average the experiment lasted less than one hour (obtained by computing the average of the time spent by the subjects which was collected from the forms), so fatigue was not very relevant. Also, the different order in which the forms were given helped to cancel out these effects.

Table 4
Spearman correlation results of the first experiment

Metric	Understandability	Analyzability	Modifiability	Understanding time
NA	0.629 $p = 0.005$	0.612 $p = 0.007$	0.675 $p = 0.002$	0.521 $p = 0.027$
NWP	0.756 $p = 0$	0.789 $p = 0$	0.784 $p = 0$	0.834 $p = 0$
NPR	0.149 $p = 0.555$	0.042 $p = 0.868$	0.148 $p = 0.558$	0.001 $p = 0.997$
NDWPIn	0.841 $p = 0$	0.830 $p = 0$	0.871 $p = 0$	0.876 $p = 0$
NDWPOut	0.802 $p = 0$	0.855 $p = 0$	0.858 $p = 0$	0.831 $p = 0$
NDWP	0.888 $p = 0$	0.892 $p = 0$	0.931 $p = 0$	0.903 $p = 0$
NDA	0.481 $p = 0.043$	0.498 $p = 0.035$	0.532 $p = 0.023$	0.425 $p = 0.079$
NCA	−0.201 $p = 0.424$	−0.254 $p = 0.308$	−0.243 $p = 0.331$	−0.190 $p = 0.451$
RDWPIn	0.220 $p = 0.381$	0.131 $p = 0.605$	0.145 $p = 0.565$	0.197 $p = 0.434$
RDWPOut	−0.220 $p = 0.381$	−0.131 $p = 0.605$	−0.145 $p = 0.565$	−0.197 $p = 0.434$
RWPA	0.189 $p = 0.453$	0.227 $p = 0.365$	0.173 $p = 0.493$	0.342 $p = 0.165$
RRPA	−0.385 $p = 0.114$	−0.454 $p = 0.059$	−0.412 $p = 0.089$	−0.421 $p = 0.082$

- *Persistence effects*. In order to avoid persistence effects, the experiment was run with subjects who had never done a similar experiment before.
- *Subject motivation*. All the professors who were involved in this experiment participated voluntarily, in order to help us in our research. We also motivated students who participated in the experiment, explaining to them that similar tasks to the experimental ones could be done in exams or practice and it could be useful in their professional career.
- Other factors. Plagiarism and influence among students could not really be controlled. Students were told that talking with each other was forbidden, but they did the experiment alone without any supervision, so we had to trust them as far as that was concerned. We are conscious that this aspect at some extent could be a threat to the validity of the experiment.

Threats to external validity. Two threats of validity have been identified which limit the possibility of applying generalization:

- *Materials and tasks used*. In the experiment we used SPMs which are representative of different standards and methodologies. Related to the tasks, the judgement of the subjects is to some extent subjective, and does not represent a real task. So more empirical studies taking "real cases" from software companies must be done.
- *Subjects*. To solve the difficulty of obtaining professional subjects, we used professors, advanced students and researchers from software engineering courses. But, we were aware that in the context of the family we had to include at least one experiment with professionals to ease the results generalization. However, the tasks to be performed with the experiments of the family, do not require high levels of industrial experience, so, experiments with students can also be appropriate.

5.2. Second experiment (replica first experiment)

In order to confirm the results obtained in the first experiment we replicated this experiment under the same conditions (strict replication) (Basili et al., 1999). As the majority of the steps are identical to those of the first experiment we will only point out those issues which were different. The subjects were fifteen professors (Software Engineering area) and ten research technicians of the Alarcos research Group of Computer Science at the Department of Computer Science at the University of Castilla-La Mancha in Spain. Hypotheses, experimental material and collected data were the same as for the first experiment. Also in this case a Kolmogorov–Smirnov test revealed that collected data was non-normal, and thus we used the Spearman's correlation coefficient, with a level of significance $\alpha = 0.05$, as reported in Table 5.

The Spearman cut-off for accepting H_0 is 0.4684 (the sample size is the same as the first experiment). Analyzing Table 5 we can conclude that there is a significant correlation between the maintainability and the metrics (NA, NWP, NDWPIn, NDWPOut, NDWP and NDA). The metric RRPA is correlated with analyzability and it seems to be less correlated with the other two maintainability sub-characteristics in comparison to the prior metrics, although it has a correlation value near to the cut-off. The metrics NPR, NCA, RDWPIn, RDWPOut and RWPA do not seem to be correlated with maintainability. With regard to the understanding time, the metrics NA, NWP, NDWPIn, NDWPOut and NDWP seem to be correlated. These results essentially confirm the results obtained in the first experiment.

6. Objective experiments

Even though the results of the subjective experiments were encouraging, we were aware that the way of mea-

Table 5
Spearman correlation results of the second experiment

Metric	Understandability	Analyzability	Modifiability	Understanding time
NA	0.684 $p = 0.002$	0.602 $p = 0.008$	0.698 $p = 0.001$	0.529 $p = 0.024$
NWP	0.724 $p = 0.001$	0.778 $p = 0$	0.750 $p = 0$	0.839 $p = 0$
NPR	0.174 $p = 0.489$	−0.012 $p = 0.962$	0.175 $p = 0.488$	0.069 $p = 0.787$
NDWPIn	0.775 $p = 0$	0.802 $p = 0$	0.847 $p = 0$	0.886 $p = 0$
NDWPOut	0.819 $p = 0$	0.854 $p = 0$	0.874 $p = 0$	0.831 $p = 0$
NDWP	0.852 $p = 0$	0.878 $p = 0$	0.917 $p = 0$	0.920 $p = 0$
NDA	0.508 $p = 0.032$	0.503 $p = 0.034$	0.558 $p = 0.016$	0.435 $p = 0.071$
NCA	−0.181 $p = 0.473$	−0.275 $p = 0.270$	−0.269 $p = 0.280$	−0.192 $p = 0.446$
RDWPIn	0.075 $p = 0.766$	0.108 $p = 0.669$	0.099 $p = 0.695$	0.208 $p = 0.408$
RDWPOut	−0.075 $p = 0.766$	−0.108 $p = 0.669$	−0.099 $p = 0.695$	−0.208 $p = 0.408$
RWPA	0.105 $p = 0.679$	0.225 $p = 0.368$	0.128 $p = 0.613$	0.334 $p = 0.176$
RRPA	−0.369 $p = 0.132$	−0.506 $p = 0.032$	−0.415 $p = 0.087$	−0.378 $p = 0.122$

suring the dependent variables was subjective and relied solely on the judgement of the users, which may have biased the results. For this reason in the family of experiments objective experiments were planned which are described in the following subsections.

6.1. Third experiment

6.1.1. Definition

The goal of the experiment is defined as: *Analyze* metrics of the SPMs structural properties *for the purpose of* evaluating *with respect to* their capability of being used as software process model maintainability indicators *from the point of view of* the researchers *in the context of* a group of software engineers of a company for the development and maintenance of information systems.

6.1.2. Planning

The planning was composed of the following activities:

Context selection. The context of the experiment was a group of professionals of a software company, and hence the experiment was run on-line (in an industrial software development environment). The subjects were thirty-one software engineers of the company Cronos Iberica Consulting, dedicated to the development and maintenance of software for information systems.

Selection of subjects. The subjects were chosen for convenience, i.e., the subjects were professionals of the software company who had wide experience and knowledge in software product modelling (UML, databases, etc.), but they did not have much experience or knowledge in the modelling of SPMs.

Variables selection. The independent variables are the SPM structural properties. The dependent variable is SPM maintainability evaluated through two of its subcharacteristics: understandability and modifiability.

Instrumentation. The objects were 18 SPMs. The dependent variables were measured by the time the subjects spent answering the questions of the first section related with the understandability of each model (understanding time) and by the time subjects spent carrying out the tasks required in the second section of the experiment (modification time). Our assumption here is that, the faster a class diagram can be understood and modified, the easier it is to maintain.

Hypothesis formulation. We wished to test the following two set of hypotheses:

- *Null hypothesis*, H_{0e}: There is no significant correlation between the metrics and the understanding time.
- *Alternative hypothesis*, H_{1e}: There is a significant correlation between the metrics and the understanding time.
- *Null hypothesis*, H_{0m}: There is no significant correlation between structural complexity metrics and the modification time.
- *Alternative hypothesis*, H_{1m}: There is a significant correlation between the metrics and the modification time.

Experiment design. We selected a within-subject design experiment. The subjects were given the forms in different order.

6.1.3. Operation

Preparation. Subjects were given a training session. They were not aware of what aspects we intended to study, neither were they aware of the actual hypothesis stated. The material we handed to the subjects consisted of the same eighteen SPMs used in the former experiments of the family and one example solved. Each diagram had an enclosed form (see Appendix B) that included two sections: the first composed of five questions related to the model and the second composed of four modification requests. Each subject had to answer the questions of Section 1 and perform the modifications specified in Section 2. For each section the subject had to specify the starting and ending understanding and modification times. The modifications to do on each SPM were similar, including the adding and deleting of activities, work products, roles and their dependences.

Execution. The subjects were given the material described in the previous paragraph. We explained how to do the forms. We allowed one week to carry out the experiment, i.e., each subject had to do the form alone. We collected all the data including the times of understanding and modification, the answers of the first section and the original models modified as a result of the second section.

Data validation. Once the data was collected, we controlled if the forms were complete and if the modifications had been done correctly. We discarded the forms of two subjects because they were incomplete. Therefore, we took into account the responses of 29 subjects.

6.1.4. Analysis and interpretation

We computed the mean of the times (understanding and modification) collected for each model. We applied the Kolmogorov–Smirnov test. As the data was non-normal we used the Spearman's correlation coefficient, with a level of significance $\alpha = 0.05$, correlating each of the metrics separately with the understanding time and modification time (see Table 6).

Because the computed Spearman's correlation coefficients for the understanding time (see Table 6) for the metrics NA, NWP, NDWPIn, NDWPOut, NDWP and NDA are above the cut-off (0.4684), and the p-value <0.05, the null hypothesis H_{0e} is rejected. Hence, we can

Table 6
Spearman correlation results of the third experiment

Metric	Understanding time	Modification time
NA	0.604 $p = 0.008$	0.171 $p = 0.496$
NWP	0.694 $p = 0.001$	0.364 $p = 0.138$
NPR	0.211 $p = 0.402$	0.348 $p = 0.157$
NDWPIn	0.740 $p = 0.000$	0.383 $p = 0.117$
NDWPOut	0.747 $p = 0.000$	0.212 $p = 0.398$
NDWP	0.772 $p = 0.000$	0.338 $p = 0.170$
NDA	0.529 $p = 0.024$	0.060 $p = 0.814$
NCA	−0.275 $p = 0.269$	0.151 $p = 0.549$
RDWPIn	0.142 $p = 0.573$	0.324 $p = 0.190$
RDWPOut	−0.142 $p = 0.573$	−0.324 $p = 0.190$
RWPA	0.150 $p = 0.554$	0.117 $p = 0.644$
RRPA	−0.304 $p = 0.220$	0.101 $p = 0.691$

conclude that there is a significant correlation between these metrics and the understanding time. With regard to modification time all the correlation values are below the cut-off and for this reason we cannot demonstrate if there is a relationship with the metrics defined. We think that these results were produced because subjects had previously answered the questions related with the understandability before performing the modification requests; this probably brought about a learning effect. This fact provided a useful feedback which was considered in the planning of the rest of experiments of the family.

6.1.5. Validity evaluation

The issues considered that could threaten the validity of this experiment were:

Threats to conclusion validity. The only issue that could affect the statistical validity of this study is the size of the sample data (522 values, 18 models and 29 subjects) but subjects were professionals which eases the generalization of the results.

Threats to construct validity. The dependent variables we used were understanding and modification times, so we consider these variables constructively valid.

Threats to internal validity. Seeing the results of the experiment we can conclude that empirical evidence of the existing relationship between the independent and dependent variables exists. We tackled different aspects that could have threaten the internal validity of the study, such as: differences among subjects, knowledge of the universe of discourse among SPMs, precision in the time values, learning effects, fatigue effects, persistence effects and subject motivation. Besides the learning effects produced, the only issue which could affect internal validity was fatigue effects because the average duration of the experiment was two hours and twenty-four minutes. But in this experiment we think it did not have a considerably affect because subjects were professionals. For the next experiments of the family with students this fact was considered.

Threats to external validity. Three threats to external validity were identified which limited the realism of the experiment (Sjoberg et al., 2002) and the ability to generalize the research results to the population under study:

- *Materials and tasks used*. In this sense, more empirical studies, using real software process models from software companies, should be carried out in future experiments or families.
- *Subjects*. The experiment was performed by professional subjects which eases the generalization of the results.
- *Environment*. The experiment was performed in the company but the tasks had to be done by using pen and paper. In future families of experiments we could consider the use of software tools to perform the activities required in order to provide a more realistic environment.

6.2. Fourth experiment

This experiment was initially planned to be a replica of the third experiment with the variation of the context variables (students as subjects) in the environment in which the solution was evaluated (Basili et al., 1999). However, as a result of the third experiment we obtained some conclusions and aspects to be improved, pointing out specially the following needs:

- To reduce the average duration of the experiment (two hours and twenty minutes in the third experiment according to the times collected from the forms), considering that the subjects of this experiment would be students, in order to avoid fatigue effects.
- To separate the activities related with the understanding of the models from the activities related with the modification. In the third experiment these two kinds of tasks had to be performed on each model which could produce a learning effect in the modification tasks that clearly affected the results with regard to the modification time.

The context of the fourth experiment of the family were two groups of students enrolled at the Department of Computer Science at the University of Castilla-La Mancha in Spain. The first group was composed of 46 students enrolled in the final-year (third) of the Computer Science BSc with a specialisation in Management and the second group were 41 students enrolled in the final-year in the Systems specialisation the Computer Science BSc. The subjects had experience and knowledge in software product modelling (UML, databases, etc.), but they did not have any experience or knowledge in

the modelling of SPMs and for this reason they were trained before the experiment took place.

The objects were 10 SPMs which are a subset of the 18 original models provided in the previous experiments of the family and they were selected to avoid the fatigue effects. The dependent and independent variables and the hypotheses were the same as for the third experiment.

In the experiment design phase we selected a within-subject design experiment. The subjects were given the forms in different order and in order to reduce the duration of the experiment we selected a representative subgroup of ten models. To provide a subgroup representative of the original 18 models, from the point of view of structural complexity, we took into account the metric values and the understanding times for each model obtained as a result of the third experiment. The models were grouped in the following way:

- Group X: Models 1, 2, 3, 9 and 10.
- Group Y: Models 4, 6, 7, 12 and 17.

For each model two different exercise forms were prepared: one in which it was required to answer the understandability questions (Xu, Yu) and another one containing the modification exercises (Xm, Ym). To separate the understandability and modifiability tasks each subject were given material composed of 10 models: five with understandability exercises and five with modifiability exercises. Subjects were arranged into two groups: Management with the models groups Xu, Ym and Systems with the material packages Yu and Xm.

The experiment took place on the same day but with a different timetable for each group of subjects (management and systems). Subjects were given an intensive training session before the experiment took place. Each model had an enclosed form as in Appendix B, but including only one of the following two sections: the first was composed of five questions related to the model and the second was composed of four modification requests. Depending on the model and the group of subjects (management or systems), each subject had to answer the questions or perform the modifications specified.

In the execution phase, we gave the subjects all the materials described in the previous paragraph. The experiment execution was controlled, as it was supervised by the researchers. We explained how to do the forms and allowed one hour and a half to carry out the experiment (we previously performed a pilot experiment to determine the average duration). We collected all the data consisting of the understanding and modification times.

Once the data were collected, we controlled if the forms were complete and if the modifications had been done correctly. We discarded the forms of one subject in the management group because the times in three models were missing. Therefore, we took into account the responses of 45 subjects in the group of management and 41 subjects in the group of systems.

We had the metric values calculated for each SPM and we calculated the mean of the understanding and modification times. As the data distribution was non-normal we decided to use the Spearman's correlation coefficient (see Table 7).

For a sample size of 10 (mean values of the times for each model) and $\alpha = 0.05$, the Spearman cut-off for accepting H_{0e} and H_{0m} is 0.6320. Because the computed Spearman's correlation coefficients for the understanding time (Table 7) for the metrics NA, NWP, NDWPIn, NDWPOut, NDWP, NDA and NCA are above the cut-off, and the p-value <0.05, the null hypothesis H_{0e}, is rejected. Hence, we can conclude that there is a significant correlation between these metrics and the understanding time. With regard to the modification time, there is a relationship between the metrics NA, NWP, NDWPIn, NDWPOut and NDWP and the time required to perform the modifications required on the models.

According to the validity evaluation the various issues that threaten the validity of the empirical study were:

- *Threats to conclusion validity*. The size of the sample data (860 values, 10 models and 86 subjects) constitutes a significant size that allowed us to obtain a conclusion validity.
- *Threats to construct validity*. The variables are constructively valid.
- *Threats to internal validity*. We tackled the aspects that could threaten the internal validity in the same way as for the previous experiments. With regard to the previous experiment with professionals, in this experiment we specially focused on the subjects motivation by giving them a special lecture about software process modelling and technology and we reduced the average duration of the experiment to avoid fatigue effects.
- *Threats to external validity*. We took into account the same factors as for the third experiment with professionals.

Table 7
Spearman correlation results of the fourth experiment

Metric	Understanding time	Modification time
NA	0.841 $p = 0.002$	0.640 $p = 0.046$
NWP	0.826 $p = 0.003$	0.650 $p = 0.042$
NPR	0.074 $p = 0.838$	0.377 $p = 0.283$
NDWPIn	0.786 $p = 0.007$	0.738 $p = 0.015$
NDWPOut	0.886 $p = 0.001$	0.791 $p = 0.006$
NDWP	0.893 $p = 0.001$	0.707 $p = 0.022$
NDA	0.821 $p = 0.003$	0.599 $p = 0.067$
NCA	−0.752 $p = 0.012$	−0.44 $p = 0.203$
RDWPIn	0.79 $p = 0.828$	0.115 $p = 0.751$
RDWPOut	−0.79 $p = 0.828$	−0.115 $p = 0.751$
RWPA	−0.116 $p = 0.751$	−0.30 $p = 0.934$
RRPA	−0.560 $p = 0.092$	−0.141 $p = 0.697$

The results of this experiment confirm the results obtained in the previous experiment because the metrics NA, NWP, NDWPIn, NDWPOut, NDWP and NDA were related to understandability. Besides, with the carrying out of this experiment it was possible to demonstrate the relationship between the metrics NA, NWP, NDWPIn, NDWPOut and NDWP and modifiability. These results confirm our conclusion at the end of the third experiment that the understandability tasks had influenced the time used by the professionals to perform the modifications required. Moreover, resulting from the fourth experiment, it seems that there could be a relationship between the metric NCA and understandability; this outcome needed to be confirmed with the replicas of the fifth experiment.

6.3. Fifth experiment (replica of the fourth experiment)

In order to confirm the results obtained in the fourth experiment we replicated this experiment with students of two Italian universities under the same conditions (strict replication). As the majority of the steps are identical, we will only point out the most significant issues.

6.3.1. First replica (University of Sannio)

The subjects were 26 undergraduate students of the Computer Engineering Laurea Degree at University of Sannio in Benevento (Italy). The experiment took place in the course of Management of Software Systems (third year).

The same sets of hypotheses as that of the fourth experiment (H_{oe}, H_{1e}, H_{om}, H_{1m}) were formulated. We gave the subjects the same material of the last experiment, which was divided in two groups (X, Y). Each subject received material composed of ten SPMs (five with understandability questions and five with modification requests). Subjects were arranged in two groups (A, B) and the material distribution was: Group A: Models X_{und}, Y_{mod}, Group B: Models Y_{und}, X_{mod}. The material also included a solved example in which it was indicated how to do the experiment and it was translated into the Italian language in order to avoid possible validity threats. Before the experiment took place we gave a training session in which the SPEM notation was explained, one example was solved and we indicated how the subjects should do the experiment.

We collected all the data obtained from the responses of the forms. Once the data was collected, we controlled if the forms were complete. Finally two subjects of Group B were discarded because their forms were not complete. We calculated the mean of the understandability and modification times and applied the Kolmogorov–Smirnov test. As the data's distribution was non-normal we used the Spearman's correlation coefficient, with a level of significance $a = 0.05$, correlating each of the metrics separately with understanding and modification times (see Table 8).

Analyzing Table 8 we can conclude that there is a correlation (rejecting H_{oe}) between the metrics NWP, NDWPIn, NDWPOut, NDWP and the understanding time. The metrics NPR, RDWPIn, RDWPOut, NCA, RWPA and RRPA do not seem to be correlated. The metrics NA and NDA have correlation values near to the cut-off. With regard to the modification time there is a relationship (rejecting H_{0m}) between the metrics NA, NWP, NDWPIn, NDWPOut NDWP and NDA. However its correlation is not demonstrated with the metrics NPR, RDWPIn, RDWPOut, RWPA and RRPA. The metric NCA has a value near to the cut-off.

The actions performed to alleviate the impact of the possible validity threats were the same as for the last experiment and in particular the translation of the material into the Italian language was considered to avoid possible misunderstandings of the subjects in doing the tasks required in the experiment. Being a replica, we consider the sample size as enough, because the objective was to confirm the results of the fourth experiment.

The results of this replica confirmed the results obtained in the fourth experiment except for the metric NA with regard to the understanding time and the metric NCA with regard to the modifiability, although their correlation values are near to the cut-off.

6.3.2. Second replica (University of Federico II)

The subjects were 38 undergraduate students of the Computer Engineering Degree at University of Federico II in Naples (Italy). The experiment took place in the course of Software Engineering (third year).

The same sets of hypotheses of the fourth experiment (H_{oe}, H_{1e}, H_{om}, H_{1m}) were formulated and we gave the subjects the same material as for the last replica. Subjects were arranged in two groups (A, B) of nineteen subjects and the material distribution was: Group A: Models X_{und}, Y_{mod}, Group B: Models Y_{und}, X_{mod}. Before the experiment took place we gave a training session with the same contents as for the last replica.

Table 8
Spearman correlation results of the fifth experiment (first replica)

Metric	Understanding time	Modification time
NA	0.555 $p = 0.096$	0.685 $p = 0.029$
NWP	0.881 $p = 0.001$	0.854 $p = 0.002$
NPR	0.253 $p = 0.480$	0.142 $p = 0.695$
NDWPIn	0.878 $p = 0.001$	0.875 $p = 0.001$
NDWPOut	0.656 $p = 0.039$	0.886 $p = 0.001$
NDWP	0.866 $p = 0.001$	0.878 $p = 0.001$
NDA	0.550 $p = 0.099$	0.647 $p = 0.043$
NCA	−0.465 $p = 0.176$	−0.506 $p = 0.136$
RDWPIn	0.479 $p = 0.162$	0.243 $p = 0.498$
RDWPOut	−0.479 $p = 0.162$	−0.243 $p = 0.498$
RWPA	0.224 $p = 0.533$	0.097 $p = 0.789$
RRPA	−0.178 $p = 0.623$	−0.357 $p = 0.311$

Table 9
Spearman correlation results of the fifth experiment (second replica)

Metric	Understanding time	Modification time
NA	0.869 $p = 0.001$	0.517 $p = 0.126$
NWP	0.701 $p = 0.024$	0.720 $p = 0.019$
NPR	0.056 $p = 0.878$	0.238 $p = 0.507$
NDWPIn	0.651 $p = 0.041$	0.719 $p = 0.019$
NDWPOut	0.794 $p = 0.006$	0.689 $p = 0.027$
NDWP	0.783 $p = 0.007$	0.648 $p = 0.043$
NDA	0.865 $p = 0.001$	0.479 $p = 0.162$
NCA	−0.798 $p = 0.006$	−0.322 $p = 0.364$
RDWPIn	−0.024 $p = 0.947$	0.231 $p = 0.521$
RDWPOut	−0.024 $p = 0.947$	−0.231 $p = 0.521$
RWPA	−0.267 $p = 0.455$	0.166 $p = 0.626$
RRPA	−0.615 $p = 0.058$	−0.080 $p = 0.826$

We collected all the data obtained from the responses of the forms and controlled if the forms were complete. All the tests were complete. We calculated the mean of the understanding and modification times for each model and we used the Spearman's correlation coefficient (the data was not-normal) (see Table 9).

According to the correlation values shown in Table 9 we can conclude that there is a correlation (rejecting H_{oe}) between the metrics NA, NWP, NDWPIn, NDWPOut, NDWP, NDA and NCA and the understanding time. The metrics NPR, RDWPIn, RDWPOut, RWPA do not seem to be correlated. The metric RRPA has correlation values near to the cut-off. With regard to the modification time there is a relationship (rejecting H_{0m}) with the metrics NWP, NDWPIn, NDWPOut and NDWP. However its correlation is not demonstrated with the metrics NPR, RDWPIn, RDWPOut, RWPA and RRPA. The metrics NA and NDA have values relatively near to the cut-off. We performed the same actions of the last replica to alleviate the impact of the possible validity threats.

The results obtained with this replica confirm widely the results obtained in the former replica and in the fourth experiment. The main differences obtained in relation to the last replica are that the metrics NA and NDA are related with the understanding time (not demonstrated in the former replica), although with this replica is not clearly demonstrated the relationship between these metrics with the modifiability (demonstrated in the former replica). Anyway, in both cases the correlation values were near to the cut-off, which in the context of the family, confirms that the metrics are valid as demonstrated in the fourth experiment.

7. Family data analysis

Once the individual experiments were carried out we performed a global analysis of the results in the context of the family of experiments to determine if the general goal of the empirical validation has been achieved. In

Table 10
Summary of the results of the experiments family

Experiments	Subjects	No. subjects	No. Mod	Dependent variables	Measurement of dependent variables	Empirically validated metrics
First	Professors, Researchers, Students	20	18	Understandability (U) Analyzability (A) Modifiability (M)	Subjective rating of subjects (U, A, M)	U, A, M: NA, NWP, NDWPIn, NDWPOut, NDPT, NDA
					Understanding time (UT)	UT: NA, NWP, NDWPIn, NDWPOut, NDPT, NDA
Second (replica of the first)	Professors, Researchers, Students	25	18	Understandability (U) Analyzability (A) Modifiability (M)	Subjective Rating of Subjects (U, A, M)	E, A, M: NA, NWP, NDWPIn, NDWPOut, NDPT, NDA, RRPA (only A)
					Understanding time (UT)	UT: NA, NWP, NDWPIn, NDWPOut, NDPT
Third	Professionals	29	18	Understandability (U) Modifiability (M)	Understanding time (UT)	UT: NA, NWP, NDWPIn, NDWPOut, NDPT, NDA
					Modification time (MT)	MT:—
Fourth	Students	86	10	Understandability (U) Modifiability (M)	Understanding time (UT)	UT: NA, NWP, NDWPIn, NDWPOut, NDPT, NDA, NCA
					Modification time (MT)	MT: NWP, NDWPIn, NDWPOut, NDPT
Fifth (replica of the fourth)	R1 Students	26	10	Understandability (U) Modifiability (M)	Understanding time (UT)	UT: NWP, NDWPIn, NDWPOut, NDPT
					Modification time (MT)	MT: NA, NWP, NDWPIn, NDWPOut, NDPT, NDA
	R2 Students	38	10	Understandability (U) Modifiability (M)	Understanding time (UT)	UT: NA, NWP, NDWPIn, NDWPOut, NDPT, NDA, NCA
					Modification time (MT)	MT: NWP, NDWPIn, NDWPOut, NDPT

Table 10 a general summary of the results obtained in the individual experiments is provided. Five experiments grouped in two subjective and three objective ones were performed in which 224 subjects participated, belonging to the following groups: students, researchers, assistant professors and professionals. According to the results the following general conclusions were obtained:

- The metrics NA, NWP, NDWPIn, NDWPOut, NDWP and NDA are valid metrics which can be used as SPMs maintainability indicators. This significant group of metrics were correlated in all the experiments with the dependent variables studied.
- The metric NCA was not validated as a result of the subjective experiments, but it is correlated with the understanding time in the fourth experiment and in the second replica in the fifth experiment. As a result, it seems that NCA could be also a useful understandability indicator, but it is necessary to confirm it with new empirical studies focused on this metric.
- Also it could be necessary to consider in future studies the metric RRPA because although it has been only correlated with the analyzability in the second experiment, its values of correlation in the majority of the experiments of the family were relatively near of the cut-off.
- The metric NPR does not seem to be correlated with maintainability. It suggests that the process roles do not have influence on the SPM view on which the metrics were defined. The results show that in this view the activities, work products and their dependences are the most influent elements in maintainability. Anyway, this metric should be significant in other views of the SPMs, as for instance, the view in which are defined the roles and their responsibilities on the work products. This issue could be considered in future studies.
- The metrics RDWPIn, RDWPOut and RPTA are not correlated with maintainability. In future studies these metrics could also be taken into account to demonstrate if they really have an influence or discard them definitely.

8. Conclusions and future works

Maintenance and evolution of software processes and their models is acquiring a growing importance in the software process community. As happens with software products, software processes evolve and consequent changes should be properly managed by organizations, in the context of effective software processes improvement programs.

The SPMs constitute the starting point in process management, and it is according to these models that processes are enacted and improved. From such a perspective, their improvement, based on project feedback, organizational policies, etc. becomes a strategic factor for meeting organization's goals. Therefore, SPMs' maintainability becomes a relevant quality factor to evaluate.

In this paper we have proposed and empirically validated a set of representative metrics to evaluate the maintainability of descriptive SPMs. These metrics are based on the main elements included in a SPM and can be used to ease SPMs evolution. In order to empirically validate the metrics proposed we carried out a family of experiments from which we obtained significant conclusions. As a result of this study, we can conclude that the metrics NA, NWP, NDWPIn, NDWPOut, NDWP and NDA are good maintainability indicators.

The metrics provide companies with objective information about the maintainability of their SPMs. More maintainable SPMs can benefit the management of the software processes in the following ways:

- A better understanding and communication of the processes which eases its later active modelling and enactment.
- More easiness to reflect the changes between the models and their enacted projects which contributes to preserving their accuracy.
- Reduction of the costs and effort necessary to change the models.

The main future lines to consider in the context of our research are:

- To carry out new families of experiments focused on the evaluation of concrete metrics we consider relevant (NPR, NCA) and that according to the results obtained in this work do not seem to be clearly correlated with the maintainability of software process models.
- To carry out case studies using real software process models.
- To consider other views related with the modelling of software processes, as for example roles and their responsibilities on work products, in order to define and validate new possible metrics.
- To develop new empirical studies to find out if the SPMs structural complexity has an influence on the project execution results.

Acknowledgments

This research is supported by the MAS project partially supported by the "Dirección General de Investi-

gación of the Ministerio de Ciencia y Tecnología'' (TIC 2003-02737-C02-02). We would also like to thank the professors, professionals of Cronos Iberica S.A and the students of Spain and Italy who participated in the experiments.

Appendix A. Subjective experiments form

Write down the time just before starting to observe the model (starting time) and the time after the rating of the model (ending time).

Starting time (indicating hh:mm:ss):__

SPM 1. With the SPM shown (Fig. 1), rate according to your criteria the following maintainability sub-characteristics:

Understandability. Easiness with which the model can be understood.

Extremely difficult	Very difficult	A bit difficult	Neither difficult nor easy	Quite easy	Very easy	Extremely easy

Analyzability. Easiness shown by the model in discovering errors or deficiencies and in guessing the parts that should be modified.

Extremely difficult	Very difficult	A bit difficult	Neither difficult nor easy	Quite easy	Very easy	Extremely easy

Modifiability. Easiness with which the model can be modified, for possible errors, a specific modification request or new requirements.

Extremely difficult	Very difficult	A bit difficult	Neither difficult nor easy	Quite easy	Very easy	Extremely easy

Ending time (indicating hh:mm:ss):__

Appendix B. Objective experiments form

SPM 1. With the SPM shown (Fig. 1), you have to perform the following tasks:

Tasks: Part I. Answer the following questions:

Write down the starting hour (indicating hh:mm:ss):__

1. Can the Technical Designer **Define the User Interface**?__
2. Is it possible to initiate the activity **Refine the User Interface** before the activity **Define the User Interface**?__
3. Is it necessary to use the product *User Work Processes* for the activity **Refine the User Interface**?__
4. Is the product *User Interface (refined)* an output of the activity **Design Process Model**?__
5. When the activity **Refine User Interface** is carried out, have the *Technical Requirements* been produced?__

Write down the ending hour (indicating hh:mm:ss):__

Tasks: Part II. Carry out the necessary modifications to satisfy the following requirements:

Write down the starting hour (indicating hh:mm:ss):__

1. It is necessary to use the product *Requirements Preliminary Information* for the execution of the activity **Define Requirements.**
2. It is not necessary to finish the activity **Define Requirements** to start the **Definition of the User Interface**, but it is necessary that the activity **Definition of the User Interface** is executed after the activity **Design the Process Model**.
3. The inclusion of the new activity **Final Review** is required after the activity **Building of the Application**. This new activity receives as input the *Application* and produces an *Approval Document*.
4. The Programmer is responsible for the **Final Review** and also participates in the Building of the Application.

Write down the ending hour (indicating hh:mm:ss):__

References

Basili, V., Rombach, H., 1988. The TAME project: towards improvement-oriented software environments. IEEE Transactions on Software Engineering 14 (6), 728–738.

Basili, V., Shull, F., Lanubile, F., 1999. Building knowledge through families of experiments. IEEE Transactions on Software Engineering 25 (4), 435–437.

Becker-Kornstaedt, U., 2000. Knowledge elicitation for descriptive software process modeling. In: Proceedings of the 22nd International Conference on Software Engineering (ICSE 2000).

Bonissone, P., 1982. A fuzzy sets based linguistic approach: theory and applications. In: Gupta, M., Sanchez, E. (Eds.), Approximate Reasoning in Decision Analysis. North-Holland Publishing Company, pp. 329–339.

Briand, L., El Emam, K., Morasca, S., 1995. Theoretical and empirical validation of software product measures. International Software Engineering Research Network, Technical Report ISERN-95-03.

Briand, L., Morasca, S., Basili, V., 1996. Property-based software engineering measurement. IEEE Transactions on Software Engineering 22 (1), 68–86.

Briand, L., Wüst, J., Lounis, H.A., 1998. Comprehensive investigation of quality factors in object-oriented designs: an Industrial Case Study. In Technical Report ISERN-98-29, International Software Engineering Research Network.

Briand, L., Arisholm, S., Counsell, F., Houdek, F., Thévenod-Fosse, P., 1999. Empirical studies of object-oriented artifacts, methods, and processes: state of the art and future directions. Empirical Software Engineering 4 (4), 387–404.

Briand, L., Bunse, C., Daly, J., 2001. A controlled experiment for evaluating quality guidelines on the maintainability of object-oriented designs. IEEE Transactions on Software Engineering 27 (6), 513–530.

Carver, J., Jaccheri, L., Morasca, S., Shull, F., 2003. Using empirical studies during software courses. Experimental Software Engineering Research Network 2001–2003, LNCS 2765, pp. 81–103.

Ciolkowski, M., Shull, F., Biffl, S., 2002. A family of experiments to investigate the influence of context on the effect of inspection techniques. In: Proceedings of the 6th International Conference on Empirical Assessment in Software Engineering (EASE), Keele, UK, pp. 48–60.

Cook, J., Wolf, A., 1999. Software process validation: quantitatively measuring the correspondence of a process to a model. ACM Transactions on Software Engineering and Methodology 8 (2), 147–176.

Curtis, B., 1992. Maintaining the software process. In: Proceedings of the International Conference on Software Maintenance. IEEE Computer Society Press, Orlando, Florida, pp. 2–8.

Curtis, B., Kellner, M., Over, J., 1992. Process modeling. Communications of ACM 35 (9), 75–80.

Dowson, M., Fernström, C., 1994. Towards requirements for enactment mechanisms. In: Proceedings of the Third European Workshop on Software Process Technology (EWSPT '94), Villard de Lans, France, LNCS 772.

Florac, W.A., Carleton, A.D., 1999. Measuring the Software Process. Statistical Process Control for Software Process Improvement. Addison Wesley.

Fuggetta, A., 2000. Software process: a roadmap. In: Proceedings of the 22nd International Conference on Software Engineering, Limerick, Ireland, pp. 25–34.

Genero, M., Piattini, M., Manso, E., Cantone, G., 2003. Building UML class diagram maintainability prediction models based on early metrics. In: Proceedings of the 9th International Symposium on Software Metrics (Metrics 2003). IEEE Computer Society, Sidney, Australia, pp. 263–275.

Godo, L., López de Mántaras, R., Sierra, C., Verdaguer, A., 1989. MILORD: the architecture and management of linguistically expressed uncertainty. International Journal of Intelligent Systems 4, 471–501.

Harrison, R., Counsell, S., Nithi, R., 2000. Experimental assessment of the effect of inheritance on the maintainability of object-oriented systems. Journal of Systems and Software 52, 173–179.

Höst, M., Regnell, B., Wholin, C., 2000. Using students as subjects—a comparative study of students and professionals in lead-time impact assessment. In: Proceedings of the 4th Conference on Empirical Assessment and Evaluation in Software Engineering (EASE), Keele University, UK, pp. 201–214.

IEEE, 1990. Institute of Electrical and Electronics Engineers. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York.

ISO/IEC, 1998. ISO/IEC 15504 TR2:1998, part 2: A reference model for processes and process capability.

ISO/IEC, 2000a. Quality Management Systems—Fundamentals and Vocabulary, ISO 9000:2000, (2000). Available from: <http://www.iso.ch/iso/en/iso9000-14000/iso9000/selection_use/iso9000family.html>.

ISO/IEC, 2000b. Quality Management Systems—Requirements ISO 9001:2000.

Jaccheri, L., Conradi, R., 1993. Techniques for process model evolution in EPOS. IEEE Transactions on Software Engineering 19 (12), 1145–1156.

Kitchenham, B., Pfleeger, S., Pickard, L., Jones, P., Hoaglin, D., El Emam, K., Rosenberg, J., 2002. Preliminary guidelines for empirical research in software engineering. IEEE Transactions on Software Engineering 28 (8), 721–734.

Li, W., Henry, S., 1993. Object-oriented metrics that predict maintainability. Journal of Systems and Software 23 (2), 111–122.

Marchesi, M., 2003. OOA Metrics for the unified modeling language. In: Proceedings of the 2nd Euromicro Conference on Software Maintenance and Reengineering, pp. 67–73.

Miranda, D., Genero, M., Piattini, M., 2003. Empirical validation of metrics for UML statechart diagrams. In: Proceedings of the 5th International Conference on Enterprise Information Systems (ICEIS 03), vol. 1, pp. 87–95.

OMG, 2002. Software Process Engineering Metamodel Specification (SPEM); adopted specification, version 1.0, Object Management Group. Available from: <http://cgi.omg.org/cgi-bin/doc?ptc/02-05-03>.

Perry, D., Porte, A., Votta, L., 2000. Empirical studies of software engineering: a roadmap. In: Proceedings of the 22nd International Conference on Software Engineering (ICSE), Future of Software Engineering Track. ACM, Limerick, Ireland, pp. 345–355.

Poels, G., Dedene, G., 2000. Distance-based software measurement: necessary and sufficient properties for software measures. Information and Software Technology 42 (1), 35–46.

Saeki, M., 2003. Embedding metrics into information system development methods: an application of method engineering technique. In: Proceedings of the 15th International Conference on Advanced Information Systems Engineering (CAiSE 2003), Lecture Notes in Computer Science, vol. 2681, pp. 374–389.

SEI, 1995. The Capability Maturity Model: Guidelines for Improving the Software Process, Software Engineering Institute (SEI). Available from: <http://www.sei.cmu.edu/cmm/cmm.html>.

SEI, 2002. Capability Maturity Model Integration (CMMI), Software Engineering Institute (SEI), version 1.1. Available from: <http://www.sei.cmu/cmmi/cmmi.html>.

Sjoberg, D., Anda, B., Arisholm, E., Dyba, T., Jorgensen, M., Karahasanovic, A., Koren, E., Vokác, M., 2002. Conducting realistic experiments in software engineering. In: Proceedings of the 2002 International Symposium on Empirical Software Engineering (ISESE'02), pp. 17–26.

Wohlin, C., Runeson, P., Höst, M., Ohlson, M., Regnell, B., Wesslén, A., 2000. Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers.

Gerardo Canfora is a full professor of computer science at the Faculty of Engineering and the Director of the Research Centre on Software Technology (RCOST) of the University of Sannio in Benevento, Italy. He serves on the program committees of a number of international conferences. He was a program co-chair of the 1997 International Workshop on Program Comprehension, of the 2001 International Conference on Software Maintenance, and of the 2004 European Conference on Software Maintenance and Reengineering; he was the General chair of the 2003 European Conference on Software Maintenance and Reengineering. His research interests include soft-

ware maintenance, program comprehension, reverse engineering, workflow management, metrics, and experimental software engineering. He serves on the Editorial Board of the IEEE Transactions on Software Engineering and The Journal of Software Maintenance and Evolution. He is a member of the IEEE and the IEEE Computer Society.

Félix García is a MSc and PhD in Computer Science from the University of Castilla-La Mancha (UCLM). Assistant Professor at the Department of Computer Science at the UCLM, in Ciudad Real (Spain). Author of several papers and book chapters on software processes management, from the point of view of their modelling, measurement and technology. He is a member of the ALARCOS research group specialized in information system quality. His research interests are: software processes and software measurement.

Mario Piattini is a MSc and PhD in Computer Science from the Politechnical University of Madrid. MSc in Psychology from the UNED. Certified Information System Auditor and Certified Information Security Manager from ISACA (Information System Audit and Control Association). Full Professor at the Department of Computer Science at the University of Castilla-La Mancha, in Ciudad Real, Spain. Author of several books and papers on databases, software engineering and information systems. He leads the ALARCOS research group specialized in information system quality. His research interests are: software quality, advanced database design, metrics, software maintenance, information system audit and security.

Francisco Ruiz is a PhD in Computer Science from the University of Castilla-La Mancha (UCLM) in 2003, and MSc in Chemistry-Physics from the University Complutense of Madrid in 1982. He is a full time associate professor of the Department of Computer Science at UCLM in Ciudad Real (Spain). He was the Dean of the Faculty of Computer Science between 1993 and 2000. Previously, he was Computer Services Director in the above-mentioned university (1985–1989) and he has also worked in private companies as an analyst-programmer and project manager. His current research interests include: software process technology and modelling, software maintenance, and methodologies for software projects planning and managing. In the past, other work topics have been: GIS (geographical information systems), educational software systems and deductive databases. He has written eight books and fourteen chapters on the mentioned topics and he has published ninety papers in national and international journals, congresses and conferences. He has been a member of nine program committees and seven organizing committees. He's a member of several scientific and professional associations (ACM, IEEE-CS, ATI, AEC, AENOR, ISO JTC1/SC7, EASST, AENUI and ACTA).

Corrado Aaron Visaggio obtained the Laurea degree in Electronic Engineering at Politecnico of Bari, Italy, in 2001. He developed his Master Thesis at the Fraunhofer IESE, Germany. In 2002 he started attending PhD courses in Software Engineering at University of Sannio, in Benevento, Italy. Currently he is working as a researcher at the Research Centre of Software Technology, RCOST, in Benevento, Italy. His main topics of research are: Software Process Modelling and Management, Agile Methodologies for developing Software, Knowledge Management in Software Engineering.