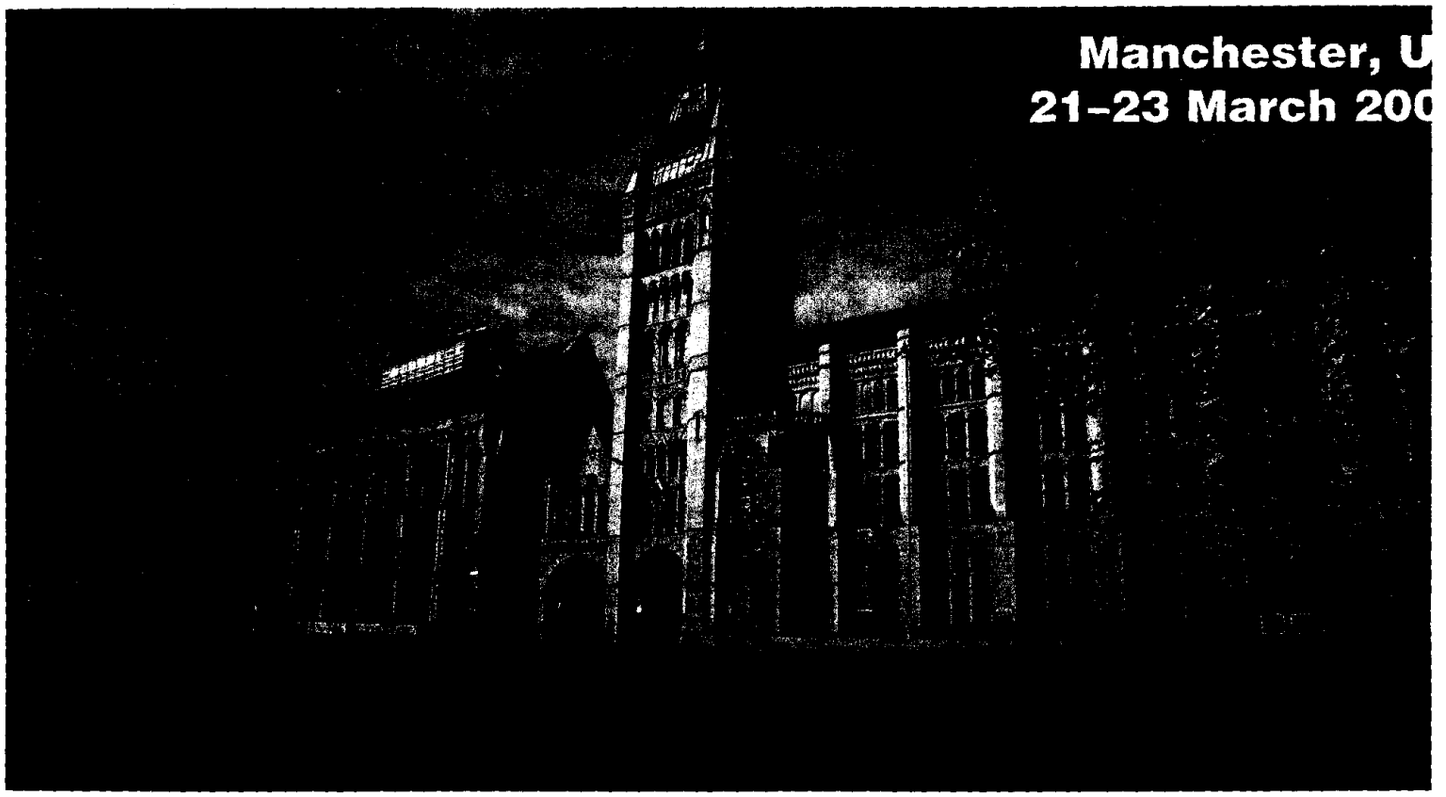
Ninth European Conference on

Software Maintenance and Reengineering



2005

Sponsored by Reegineering Forum (REF)
IEEE CS Technical Council on Software Engineering (TCSE)
University of Manchester, School of Informatics
In Cooperation with The British Computer Society (BCS)
King's College London

IEEE COMPUTER SOCIETY

IEEE

Reengineering Forum

The University of Manchester School of Informatics

BCS® THE BRITISH COMPUTER SOCIETY

KING'S College LONDON

Proceedings

Ninth European Conference on Software Maintenance and Reengineering

CSMR

Proceedings

Ninth European Conference on Software Maintenance and Reengineering

21 - 23 March 2005
Manchester, UK

Sponsored by
Reengineering Forum (REF)
IEEE CS Technical Council on Software Engineering (TCSE)
University of Manchester, School of Informatics

In-Cooperation with
The British Computer Society (BCS)
King's College London

IEEE COMPUTER SOCIETY
IEEE

Los Alamitos, California
Washington • Brussels • Tokyo

IEEE Computer Society Order Number P2304
ISBN 0-7695-2304-8
ISSN 1534-5351

Additional copies may be ordered from:

IEEE Computer Society
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: + 1 800 272 6657
Fax: + 1 714 821 4641
http://computer.org/cspress
csbooks@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1 732 981 0060
Fax: + 1 732 981 9667
http://shop.ieee.org/store/
customer-service@ieee.org

IEEE Computer Society
Asia/Pacific Office
Watanabe Bldg., 1-4-2
Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN
Tel: + 81 3 3408 3118
Fax: + 81 3 3408 3553
tokyo.ofc@computer.org

Individual paper REPRINTS may be ordered at: reprints@computer.org

Editorial production by JD Cantarella

Cover art production by Joseph Daigle

Printed in the United States of America by The Printing House

IEEE Computer Society
Conference Publishing Services
http://www.computer.org/proceedings/

Table of Contents

Technical Session 5: Dynamic Analysis

Keynote

Technical Session 6: Program Analysis

Tool Demonstration

Technical Session 7: Migrating Legacy Systems

Short Paper Session

Maintainability of Software Process Models: An Empirical Study

F. García, M. Piattini, F. Ruiz
Alarcos Research Group
University of Castilla-La Mancha
Paseo de la Universidad, 4, 13071, Ciudad Real, Spain
{Felix.Garcia, Francisco.RuizG, Mario.Piattini}@uclm.es,
http://alarcos.inf-cr.uclm.es/english/

C.A. Visaggio
RCOST - Research Centre on Software Technology
Dipartimento di Ingegneria
Università del Sannio
Palazzo ex Poste, viale Traiano, 82100 Benevento, Italia
visaggio@unisannio.it
http://www.rcost.unisannio.it

Abstract

Adequate process modeling is one of the key factors for the success of software organizations. Currently, organizations face with a very high competition and consequently they have to continuously improve their processes. This improvement could require changes of the process models so it is important to evaluate the maintainability of these models to facilitate their evolution. In this paper we present the results obtained with the replication of an experiment to validate a set of metrics for Software Process Models (SPMs). The replicas were performed at the Italian Universities of Sannio (Italy) and Federico II (Naples). They are part of a family of experiments and have confirmed the results obtained in the original experiment carried out at the University of Castilla-La Mancha (Spain). As a result a set of useful indicators of the understandability and modifiability of the SPMs, which are two key sub-characteristics of their maintainability, have been obtained.

1. Introduction

Currently, the software processes have turned into a very important factor to consider for the success of software organizations. They have to change their processes in order to keep high competitiveness in the market. Causes of such frequent and relevant modifications could be [12; 13; 21]:

- Introduction of new technologies.
- Improvement of maturity according to models as CMM (Capability Maturity Model) [19], CMMI (Capability Maturity Model Integration) [20] and ISO 15939 [11].
- Innovative production methods: software components based [18], software product lines [22], development based on open-source software [14], agile methodologies [9].

The improvement of software processes involves the need to effectively maintain them and the maintenance of the software process deserves the same attention as well as any other kind of software [5]. Considering that "software processes are software too" [16], the evaluation of the maintainability of the processes in early stages of their development and specially, in the modeling stage, is fundamental. Software process models (SPMs) constitute the starting point to carry out the later enactment, evaluation and improvement. Therefore, as the software processes change, process models may change accordingly. Taking into account the main uses of SPMs [6] it is necessary to maintain effectively the process models with the aim to facilitate: the communication of process modifications, the understanding of new responsibilities and procedures and the automation of guidance in performing new activities.

According to the issues previously identified, our main research goal consists of providing a set of objective indicators in order to evaluate the maintainability of descriptive SPMs. It could provide companies with a quantitative basis to choose, among alternative SPMs which satisfy the business goals, the model which can be more easily maintained. To reach this goal we have quantified descriptive SPMs by means of the definition of a set of suitable metrics and we have carried out a family of experiments to find the metrics that can be used as maintainability indicators.

In this paper we describe the results obtained with the two final replicas of the family of the experiments carried out. The paper is organized as follows. Section 2 provides an overview of the overall empirical study performed previously. In Section 3 the replicas which constitute the main contribution of this paper are

1534-5351/05 $20.00 © 2005 IEEE

described in detail. Section 4 provides an analysis of the results obtained in the replicas which are compared with the results obtained in the original experiment. Finally the conclusions and future works are outlined in Section 5.

2. Overview of the Empirical Study

With the aim to provide the quantitative basis necessary to know the maintainability of the SPMs a set of metrics (Table 1) have been defined:

Table 1. Metrics of SPMs

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** $NDWP(PM) = NDWPIn(MP) + NDWPOut(MP)$
NDA	Number of precedence dependences between **Activities**
NCA	Activity Coupling in the process model. $NCA(PM) = \frac{NA(PM)}{NDA(PM)}$
RDWPIn	Ratio between **input dependences** of Work Products with Activities and **total number of dependences** of Work Products with Activities $RDWPIn(PM) = \frac{NDWPIn(PM)}{NDWP(PM)}$
RDWPOut	Ratio between **output dependences** of Work Products with Activities and **total number of dependences** of Work Products with Activities $RDWPOut(PM) = \frac{NDWPOut(PM)}{NDWP(PM)}$
RWPA	Ratio of **Work Products** and **Activities**. Average of the work products and the activities of the process model. $RWPA(PM) = \frac{NWP(PM)}{NA(PM)}$
RRPA	Ratio of **Process Roles** and **Activities** $RRPA(PM) = \frac{NPR(PM)}{NA(PM)}$

The metrics are model scope because they evaluate the overall structural complexity of a SPM and they have been defined following the SPEM (Software Process Engineering Metamodel) terminology [15] by examining its key software process constructors. Due to the generality of SPEM the metrics can be directly applied to other Process Modeling Languages (PMLs).

The objective of the research is to find a set of useful indicators of the SPMs easiness of maintenance. The metrics (Table 1) evaluate the structural complexity (internal attribute) [10] of the SPMs and once the metrics were defined, the relationship between the structural complexity and the maintainability (external attribute) was investigated. This situation is illustrated in the Figure 1, in which the metrics are classified according to the aspect of the structural complexity they capture following the Briand et al. framework [3]:

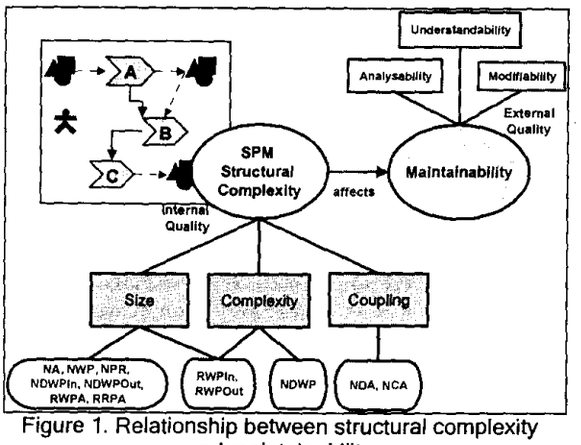


Figure 1. Relationship between structural complexity and maintainability

With the aim to establish which metrics are useful SPMs maintainability indicators, a family of experiments was carried out [4]. The experiments of the family can be classified in the following two groups:

- **Subjective Experiments.** In these two experiments the subjects (students, researchers and assistant professors) rated each of three maintainability sub-characteristics (understandability, analysability, modifiability) on a material composed of 18 SPM according to a scale composed of seven linguistic labels (from extremely easy to extremely difficult for each sub-characteristic). The results obtained in these experiments [4] reflect that several of the model level metrics (NA, NWP, NDWPIn, NDWPOut, NDWP and NDA) were highly related to software process models maintainability.
- **Objective Experiments.** Even though the results of the subjective experiments were good, we were aware that the way of measuring the dependent variables was subjective and relied solely on the judgment of the users, which may have biased the results. Therefore, we decided to carry out two additional experiments in which we measured the dependent variable in a more objective way. In these experiments the dependent variables considered were the understandability and the

modifiability of the SPMs. To measure these variables the subjects had to answer five questions and perform four modifications on the models. We obtained the time the subjects spent answering the questions (understandability time) and the time subjects spent carrying out the tasks required (modifiability time) in order to demonstrate if there was a relationship between the metrics and the understandability and modifiability times. In the first experiment of this type [7] the subjects were professionals of a software company and the material was composed of 18 models which included two sections (understandability and modifiability). As a result we could validate some metrics (NA, NWP, NDWPIn, NDWPOut, NDWP and NDA) with respect to the understandability time, but we could not demonstrate any relationship between the metrics and the modifiability time. The reason was that the subjects had to answer the questions before performing the modifications and it produced learning effects in the modifiability tasks. This fact was taken into account in the planning of the following experiment [8] in which the subjects were 87 undergraduate students of the University of Castilla-La Mancha in Spain. The material was reduced to 10 models which included only one section (understandability or modifiability). In total, five models included understandability questions and the other five the modification exercises. In this experiment the metrics NA, NWP, NDWPIn, NDWPOut, NDWP, NDA and NCA were confirmed as good understandability indicators and the metrics NA, NWP, NDWPIn, NDWPOut and NDWP as good modifiability indicators.

In order to confirm the results obtained in the last experiment we replicated it with students of two Italian universities under the same conditions (strict replication [1]). These strict replicas are described in the following section.

3. Experimental Replicas

These replicas constitute the latest empirical study performed in the context of the family of experiments carried out to find useful SPMs maintainability indicators.

3.1. Research Objectives

Using the GQM template [2] the goal of the experiment is defined as: *Analyse* SPM structural complexity metrics *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* Computer Science undergraduate students.

3.2. Experimental Design

A general view of the experimental design is provided in Figure 2:

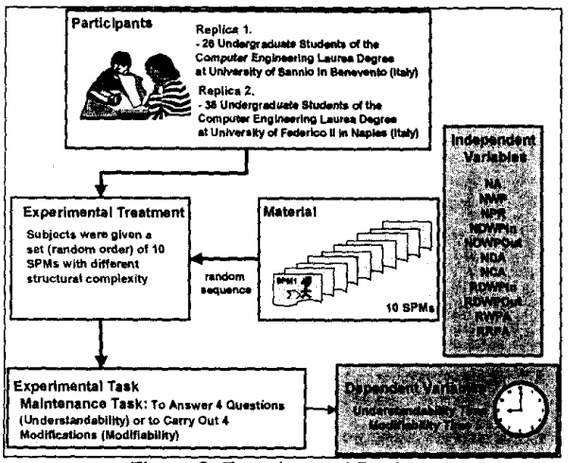


Figure 2. Experimental Design

A within-subjects design was used, in which each participant was given a material composed of ten SPMs with different levels of complexity. The order of the models provided was randomized for each subject. Five models contained five questions related to the understandability of the model and the other five contained four modifications requests. The subjects had to answer the questions (yes/no) and carry out the modifications by drawing on the original model.

3.3. Participants

The participants were students of two Italian universities. The first replica took place at the university of Sannio in Benevento (Italy) and 26 students of the third course of the Computer Engineering Laurea Degree participated. The second replica was run at the University of Federico II in Naples (Italy) and 38 students of the third course of the Computer Engineering Laurea Degree were involved.

The subjects in the two replicas had wide knowledge in software product modelling (UML, databases, etc.), but they had not experience or knowledge in the modeling of SPMs. An introductory lecture about the SPM modeling and the SPEM metamodel and notation was given and a training session was developed to provide subjects the necessary knowledge to do the tasks required in the experiment. However, the subjects were not aware of what aspects we intended to study.

3.4. Material

The experimental material was composed of ten SPMs with different values of structural complexity obtained by varying the metric values (see Table 2). The models were based in different methodologies and SPMs found in the literature, like for example PMBOK, Rational Unified Process, etc. 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 sheets were prepared: one in which it was required to answer the understandability questions (X_u, Y_u) and another one containing the modification exercises (X_m, Y_m). The material also included an example solved in which it was indicated how to do the experiment and it was translated to the Italian language by the Italian native authors in order to avoid possible validity threats. One of the sheets of the material is shown in Appendix A.

Table 2. Metric Values of the SPMs of the Replicas

Model	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
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
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
12	2	8	3	6	4	10	1	2.000	0.600	0.400	4.000	1.500
17	4	24	1	20	11	31	3	1.333	0.645	0.355	6.000	0.250

3.5. Experimental Task

Each subject received a material composed of ten SPMs (five with understandability questions and five with modification requests). Depending on the model (group X or Y) the subjects had to do one of the following tasks: to answer "yes" or "no" to five questions about the model or to carry out four modifications consisting of adding and/or deleting activities, work products, roles or dependences among these elements. In both replicas, the subjects were arranged in two groups (A, B) and the material distribution was: Group A: Models X_u, Y_m, Group B: Models Y_u, X_m.

The tasks of each type (understandability or modifiability) to develop were similar in complexity. For this reason, the only source of variation in effort to perform the tasks of the same type should be the complexity of each model. Before starting to perform the tasks required in each model the subjects had to write down the starting time and at the end they had to write down the finishing time. One hour and a half was allowed in the two replicas to carry out the experimental task. It was the same time allowed in the previous experiment which was quite enough.

3.6. Variables

The independent variable was the SPM structural complexity (*latent* independent variable) which was measured through the twelve metrics defined (*observed* independent variables). The *levels* of the independent variable are embodied in the different experimental SPMs which represent a different combination of metric values (Table 2).

The dependent variable was the SPM maintainability (easiness of maintenance of a SPM) evaluated by two of its sub-characteristics: understandability and modifiability. These dependent variables were measured by the time the subjects spent answering the questions (understandability time) and by the time subjects spent carrying out the tasks required (modifiability time). Our assumption here is that, the faster a SMP can be understood and modified, the easier it is to maintain.

Therefore, all the measures (of the independent and dependent variables) were quantitative and objective.

3.7. Hypothesis

According to the goal of our research, we wished to test the following two sets of hypotheses:

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

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

3.8. Results

Once the data were collected, we checked if the tests were complete. Finally, in the first replica two subjects of the Group B were discarded because their tests were not complete whereas in the second replica all the tests were complete. In Table 3 the mean and standard deviation of the times required to understand and modify each model are shown:

Table 3. Mean and Standard Deviation of the Understandability and Modifiability Times

SPM	Replica 1				Replica 2			
	Und Time		Mod time		Und Time		Mod Time	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
1	117	53	255	75	132	51	268	43
2	80	39	255	73	97	36	290	71
3	141	92	260	39	124	50	291	90
4	244	111	418	183	202	39	462	216
6	139	70	288	74	114	49	355	92
7	170	69	312	138	224	138	383	160
9	124	66	345	142	142	62	396	162
10	316	206	537	295	282	124	510	141
12	108	25	271	114	95	27	396	132
17	128	48	293	116	124	83	378	166

As we can observe in Table 3 the models 4, 7 and 10 were the most difficult to maintain according to their mean times. These models also show the highest standard deviation. Considering the metric values of the SPMs (Table 2) the models 4 and 10 seem to be the models with the highest structural complexity which provides some evidence about the influence of the structural complexity of the SPMs in their maintainability.

To apply the statistical analysis a summary of the data was performed. The summary was composed of the mean of the understandability and modifiability times and the values of the metrics grouped by SPMs. To choose the statistical test to use, we apply the Kolmogorov-Smirnov test and as result the distribution of the data collected was non-normal. As the data were non-normal we decided to use the Spearman's correlation coefficient, with a level of significance α= 0.05, correlating each of the metrics separately with the understandability and modifiability times. In Table 4 the correlation results obtained in the two replicas are shown:

Table 4. Spearman Correlation Results

	Replica 1		Replica 2	
Metric	Ut	Mt	Ut	Mt
NA	0.555 p=0.096	**0.685** p=0.029	**0.869** p=0.001	0.517 p=0.126
NWP	**0.881** P=0.001	**0.854** p=0.002	**0.701** p=0.024	**0.720** p=0.019
NPR	0.253 p= 0.480	0.142 p=0.695	0.056 p= 0.878	0.238 p=0.507
NDWPIn	**0.878** p=0.001	**0.875** p=0.001	**0.651** p=0.041	**0.719** p=0.019
NDWPOut	**0.656** p=0.039	**0.886** p=0.001	**0.794** p=0.006	**0.689** p=0.027
NDWP	**0.866** p=0.001	**0.878** p=0.001	**0.783** p=0.007	**0.648** **p=0.043**
NDA	0.550 p=0.099	**0.647** p=0.043	**0.865** p=0.001	0.479 p=0.162
NCA	-0.465 p=0.176	-0.506 p=0.136	-0.798 p=0.006	-0.322 p=0.364
RDWPIn	0.479 p=0.162	0.243 p=0.498	-0.024 p=0.947	0.231 p=0.521
RDWPOut	-0.479 p=0.162	-0.243 p=0.498	-0.024 p=0.947	-0.231 p=0.521
RWPA	0.224 p=0.533	0.097 P=0.789	-0.267 p=0.455	0.166 P=0.626
RRPA	-0.178 p=0.623	-0.357 p=0.311	-0.615 p=0.058	-0.080 P=0.826

For a sample size of 10 and α = 0.05, the Spearman cutoff for accepting H_{0e} and H_{0m} is 0,6320.

With respect to the understandability time, the metrics NWP, NDWPIn, NDWPOut and NDWP are correlated (rejecting the H_{0e} hypothesis) in the first replica and the metrics NA, NWP, NDWPIn, NDWPOut, NDWP and NDA in the second replica. The metrics NA and NDA in the first replica are quite near to the cut-off.

In relation to the modifiability time, in the first replica the metrics (rejecting H_{0m}) NA, NWP,

NDWPIn, NDWPOut, NDWP and NDA are correlated and in the second replica the metrics correlated are NWP, NDWPIn, NDWPOut, and NDWP being near to the cut-off the metrics NA and NDA.

According to the results obtained in the replicas the metrics NA, NWP, NDWPIn, NDWPOut, NDWP and NDA seem to be correlated with the understandability and modifiability of the SPMs.

3.9. Validity Threats

The main issues that threaten the validity of the empirical study were:

- **Internal Validity**. The following variables were controlled as part of the experiment:
  - *Participant characteristics*: the use of a within-subjects design minimized the possible threat of differences among subjects.
  - *Task complexity*: the experimental tasks were equivalent in complexity for each group of experimental models (understandability and modifiability).
  - *Instrumentation*: the same measurement techniques were used for independent and dependent variables for all participants. The risk of measurement error was reduced by calculating the values for all values automatically.
  - *Training*: all participants were given the same prior training session and they received the background necessary to carry out the experiment properly.
  - *Learning effects*: experimental models were given to subjects in random order and only one type of task (understanding or modification) was required for each model to minimize learning and sequence effects.
  - *Control of environment*: This fact did not affect the internal validity because the replicas were conducted under controlled conditions being the participants supervised by the experimenters in the classroom.
  - *Fatigue Effects*: The average duration of the replicas was of forty minutes and as a result fatigue effects were avoided.
  - *Measurement error*: Another threat to internal validity is the fact that subjects were responsible for recording the time it took to perform the experimental tasks. This increases risk of measurement error for the dependent variable, as subjects may have recorded the time inaccurately. The within-subject design helped to minimize this threat because the possible measurement error should be randomly distributed across levels of the independent variable. Besides, a digital clock was displayed during the execution of the replicas to ease participants to write down accurate times.
- **External Validity**. We identify three possible threats to the external validity of this study:
  - *Sample population*: A clear threat to the generality of the findings of this study was the type of experimental subjects. The population selected was of students which reduces the possibility to generalize the results in practice. Anyway, it has not been a serious threat because these replicas are part of the latest experiment of the family in which previously a similar objective experiment had been carried out with professionals [7]. The main goal of the replicas has been to confirm the results obtained with the previous objective experiments of the family. Besides, the tasks to be performed did not require high levels of industrial experience, and so, experiments with students could be appropriate [1].
  - *Experimental models*: In the experiment, we have used software process models based on standards and methodologies found in the literature and tasks representative of real cases, but more empirical studies, using real software process models from software companies, must be carried out.
  - *Experimental task*: The types of task to perform on the models were designed in order to accomplish the goals of the research and they should be adapted to situations in practice. With respect to the environment, the replicas were done by using pen and paper. In future experiments we could consider the use of software tools to perform the activities required in order to provide a more realistic environment.

4. Analysis of the Results

In the Table 5 the metrics which were validated in the experiment carried out at the University of Castilla-La Mancha in Spain and in the replicas in Italy are shown:

Table 5. Metrics Validated in the Objective Experiments

		NA	NWP	NDWPIn	NDWPOut	NDWP	NDA	NCA
Understandability	UCLM (Spain)	X	X	X	X	X	X	X
	Sannio (Italy)		X	X	X	X		
	Federico II (Italy)	X	X	X	X	X	X	
Modifiability	UCLM (Spain)	X	X	X	X	X		
	Sannio (Italy)	X	X	X	X	X	X	
	Federico II (Italy)		X	X	X	X		

As we can see in Table 5, the metrics NWP, NDWPIn, NDWPOut and NDWP were correlated with the understandability and modifiability in all the experiments. The metrics NA was clearly correlated in the experiment in Spain and in the cases in which it was not correlated (understandability in Sannio and modifiability in Naples) the values were near to the cut-off which confirms that it could be also a significant maintainability indicator. Other metric that seems to be significant is NDA although its correlation is not as clear as the rest of metrics. The metric NCA was correlated in the experiment in Spain but this result was not confirmed with the replicas.

The main differences obtained in the replicas are that the metrics NA and NDA at University Federico II were related with the understandability time (not demonstrated in the replica at Sannio), although with this replica was not clearly demonstrated the relationship of these metrics with the modifiability (demonstrated in the replica at Sannio). 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 4th experiment.

In Figures 3 and 4 a comparison of the average times the subjects employed to understand and modify each model in the experiments in Spain and Italy are shown:

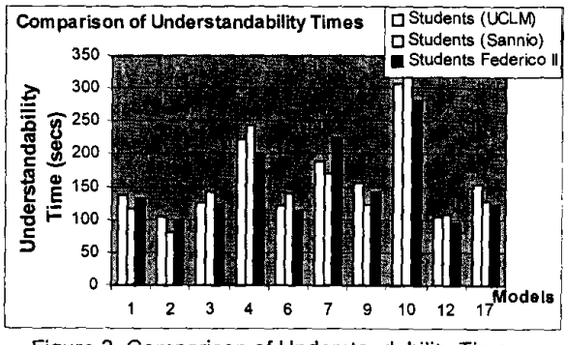


Figure 3. Comparison of Understandability Times

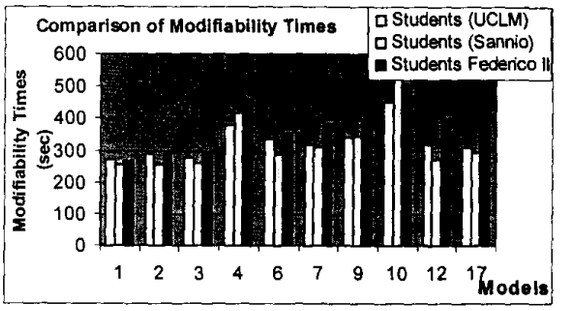


Figure 4. Comparison of Modifiability Times

As we can observe in the graphics the there were not significant differences in the times employed by the students of Spain and Italy. In average it seems that the students of the University of Federico II spent more time in the modification of the SPMs and less time in the understanding of the models than the students of UCLM and Sannio. The graphics also provide an important overview of the maintainability of each model which demonstrates that the models with less easiness of maintenance are the models which higher value of the valid metrics (NA, NWP, NDWPIn, NDWPOut, NDWP and NDA). In the future it could be important to evaluate the relative effect of the metrics in determining the SPM maintenance effort by developing prediction models.

5. Conclusions and Further Work

Maintenance is acquiring growing importance in the software process community. As it happens with software products, software processes evolve and consequent changes should be properly managed by organization, in the context of effective software processes improvement programs. Therefore, SPMs' maintainability becomes a relevant quality factor to evaluate.

In this paper we have presented the results obtained with the replication of an experiment to validate a set of metrics for SPMs. The replicas are part of a family

of experiments and have confirmed the results obtained in the original experiment carried out at the University of Castilla-La Mancha in Spain. As a result, the metrics NA, NWP, NDWPIn, NDWPOut, NDWP and NDA seem to be useful indicators of the understandability and modifiability of the SPMs, two key sub-characteristics of the maintainability. These metrics can be very useful to select the models with the most easiness of maintenance among various alternatives in companies with change their SPMs to improve their software processes. It can help to facilitate the software processes evolution in these companies by assessing the process improvement at conceptual level.

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:

- To guarantee the understanding and the diffusion of the processes, as they evolve, without affecting their successful execution.
- To reduce the effort necessary to change the models with the consequent reduction of the maintenance costs given their influence in the software lifecycle cost [17].

The results obtained with the overall family of experiments are encouraging and have allowed us to select a set of useful maintainability indicators. However, it is necessary to develop new empirical studies to confirm the usefulness of the empirically validated metrics and obtain insight enough to discard the metrics that do not have influence on the maintainability of SPMs. The lines for improvement in future studies are:

- Carrying out new experiments focused on the evaluation of concrete metrics we consider relevant (NPR, NCA) and that according to the family experiment results seem not to be clearly correlated with the maintainability of SPMs.
- Building of prediction models of the maintainability of the models by researching the concrete influence of each validated metric in the easiness of maintenance of the SPMs.
- Performing case studies using real software process models.
- Consideration of other views related with the modeling of software processes in order to define and validate new possible metrics. For example the view with the roles and their responsibilities on work products could be considered.

Acknowledgements

This research is supported by the MAS project partially supported by "Dirección General de Investigación of the Ministerio de Ciencia y Tecnología" (TIC 2003-02737-C02-02). We would also like to acknowledge the professor Giuseppe Di Luca for his help in the replica carried out at the University of Federico II in Naples (Italy).

References

[1] V. Basili, F. Shull, F. Lanubile. "Building Knowledge through Families of Experiments", *IEEE Transactions on Software Engineering*, 25(4), 1999, pp. 435-437.

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

[3] L. Briand, S. Morasca, V. Basili. "Property-Based Software Engineering Measurement", *IEEE Transactions on Software Engineering*, 22(1), 1996, pp. 68-86.

[4] G. Canfora, F. Garcia, M. Piattini, F. Ruiz, A. Vissagio. "A Family of Experiments to Validate Metrics for Software Process Models". *Journal Systems and Software* (2nd review submitted), 2004.

[5] B. Curtis. "Maintaining the Software Process", *Proceedings of the International Conference on Software Maintenance (ICSM)*, IEEE Computer Society Press, Orlando, Florida, 1992, pp. 2-8.

[6] B. Curtis, M. Kellner, J. Over. "Process Modeling", *Communications of ACM*, 35(9), 1992, pp. 75-80.

[7] F. García, F. Ruiz, M. Piattini. "Definition and Empirical Validation of Metrics for Software Process Models". *Proceedings of the 5th International Conference Product Focused Software Process Improvement (PROFES'2004)*, Lecture Notes in Computer Science (LNCS 3009), Kansai Science City (Japan), 2004, pp. 146-158.

[8] F. García, F. Ruiz, M. Piattini. "An Experimental Replica to Validate a set of Metrics for Software

Process Models", *Proceedings of the European Software Process Improvement Conference* (EuroSPI 2004), Lecture Notes in Computer Science, Trondheim (Noruega), Springer-Verlag, 2004.

[9] J. Gomez. "Balancing Productivity and Process Improvement with Agile-Based Methods: A Real Experience ", *Proceedings of the Second International Conference on Software Process Improvement (ICSPI)*, 2004.

[10] ISO/IEC 9126-1.2, "Information technology-Software product quality – Part 1: Quality model", 2001.

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

[12] D. Johnson, J. Brodman. "Applying CMM Project Planning Practices to Diverse Environments", IEEE Software, 17(4), 2000, pp. 40-47.

[13] M. Morisio. "Diversity in Reuse Processes", IEEE Software, 17(4), 2000, pp. 56-63.

[14] J. Norris, P. Camp. "Mission-Critical Development with Open Source Software: Lessons Learned", IEEE Software, 21(1), 2004, pp. 42-49.

[15] Object Management Group (OMG). "Software Process Engineering Metamodel Specification (SPEM)", adopted specification, version 1.0, 2002, available from http://cgi.omg.org/cgi-bin/doc?ptc/02-05-03

[16] L. J. Osterweil. "Software Process are Software Too, Revisited", *Proceedings of the 19th International Conference on Software Engineering (ICSE)*, Boston, MA, ACM Press, 1997, pp. 540-548.

[17] T.M Pigoski. *Practical Software Maintenance. Best Practices for Managing your Investment*, John Wiley & Sons, 1997.

[18] C. Paul. "From subroutines to Subsystems: Component-Based Software Development", *in Component-Based Software Engineering: Selected Papers from the Software Engineering Institute*, IEEE Computer Society Press, 1996.

[19] Software Engineering Institute (SEI). The Capability Maturity Model: Guidelines for Improving the Software Process, (1995). In http://www.sei.cmu.edu/cmm/cmm.html

[20] Software Engineering Institute (SEI). Capability Maturity Model Integration (CMMI[SM]), version 1.1. March (2002). In http://www.sei.cmu/cmmi/cmmi.html

[21] S. M. Sutton. "The Role of Process in a Software Start-up", IEEE Software, 17(4), 2000, pp. 33-39.

[22] D. Weiss, C. T. Lai. *Software Product-Line Engineering: A Family-Based Software Development Process*, Addison-Wesley, 1999.

Appendix A

Group A

SPM 1 (Figure 5). 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): ___

Group B

SPM 1 (Figure 5). 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 for the **Definition of the User Interface** to be executed after the activity **Design the Process Model**.

3. It is desired to include the new activity **Final Review**, after the **Building of the Application.** This new activity receives as input the *Application* and produces an *Approval Document.*
4. The Programmer is responsible of the **Final Review** and also participates in the Building of the Application.

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

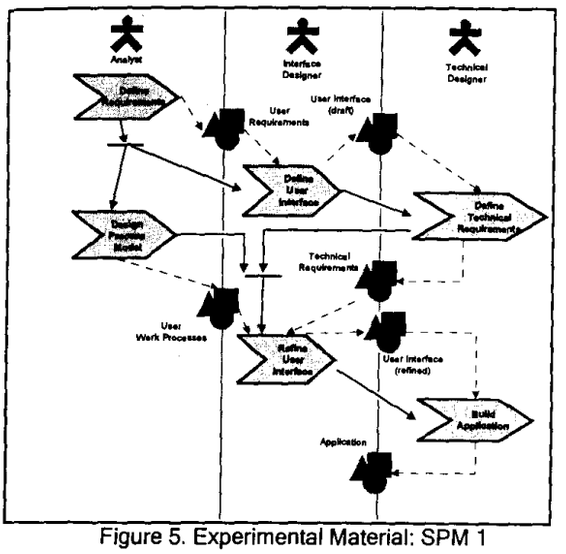


Figure 5. Experimental Material: SPM 1