9th International Software Metrics Symposium

3-5 September 2003
Sydney, Australia

Proceedings

Ninth International Software Metrics Symposium

September 3-5, 2003
Sydney, Australia

Sponsored by
IEEE Computer Society Technical Council on Software Engineering

In collaboration with
University of New South Wales, Australia
University of Technology, Sydney, Australia
National ICT, Australia
Australian Software Metrics Association

IEEE COMPUTER SOCIETY
http://computer.org
Los Alamitos, California
Washington • Brussels • Tokyo

IEEE Computer Society Order Number PR01987
ISBN 0-7695-1987-3
ISSN 1530-1435

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: + 1-714-821-8380
Fax: + 1-714-821-4641
E-mail: cs.books@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 Bob Werner
Cover art production by Joe Daigle/Studio Productions
Printed in the United States of America by The Printing House

Building UML Class Diagram Maintainability Prediction Models Based on Early Metrics

Marcela Genero, Mario Piattini
Department of Computer Science
University of Castilla-La Mancha
Ciudad Real – Spain
{Marcela.Genero, Mario.Piattini}@uclm.es

Esperanza Manso
Department of Computer Science
University of Valladolid
Valladolid - Spain
manso@infor.uva.es

Giovanni Cantone
Dipartimento di Informatica, Sistemi e Produzione
Università degli Studi di Roma "Tor Vergata"
Rome - Italy
cantone@info.uniroma2.it

Abstract

The fact that the usage of metrics in the analysis and design of object oriented (OO) software can help designers make better decisions is gaining relevance in software measurement arena. Moreover, the necessity of having early indicators of external quality attributes, such as maintainability, based on early metrics is growing. In addition to this, the aim of the present paper is to show how early metrics which measure internal attributes, such as structural complexity and size of UML class diagrams, can be used as early class diagram maintainability indicators. For this purpose, we present a controlled experiment and its replication, which we carried out to gather the empirical data which in turn is the basis of the current study. From the results obtained, it seems that there is a reasonable chance that useful class diagram maintainability models could be built based on early metrics. Despite this fact, more empirical studies, especially using data taken form real projects performed in industrial settings, are needed in order to obtain a comprehensive body of knowledge and experience.

Keywords: maintainability, class diagrams, structural complexity, size, object-oriented metrics, empirical validation, controlled experiments, prediction model.

1. Introduction

Nowadays, the idea that "measuring quality is the key to developing high-quality OO software" is gaining relevance [1]. Moreover, in the software engineering arena, it is widely recognised that, in order to obtain better quality OO software products should be on measuring the quality characteristics of early artefacts, including products that are delivered in the very beginning of OO software analysis and design [1], [2], [3], [4], [5], [6], [7], [8]. The class diagram of Key Abstractions (KA) for the general view, the View of Participating Classes (VOPC), for each use case and the VOPC union are the key UML artefacts of software analysis, while other class diagrams are the key artefacts of software design. The quality of such early artefacts has a great impact on the quality of the software product which is finally delivered. Hence, the evaluation (and improvement if necessary) of class diagram quality is a crucial issue. In this arena, software measurement plays an important role because the early availability of metrics contributes objectively to class diagram quality evaulation, avoiding bias in the quality evaluation process.

We have focused on class diagram maintainability. This is one of the main software product quality characteristics [9]. Concerns with software development organizations are mostly related to this characteristic; in fact, maintenance is still the major resource consumer of the whole software life cycle [10]. However, because maintainability is an "external quality attribute", it can be evaluated only when the product is nearly or completely finished. Consequently, in order to to make an early evaluation of such qualities, it is necessary to make further indicators available. These indicators should be based on properties of early artefacts, e.g., the structural properties of class diagrams [4], and their structural complexity and size.

After a thorough review of some of the existing OO measures that can be applied to class diagrams at the initial phases of OO software development life cycle, we have proposed a set of 8 measures for UML class diagram structural complexity [11], [12], [13][1], related to the usage of UML relationships, such as associations, dependencies, aggregations and generalizations[2]. In this study, we also consider traditional OO metrics, such as size metrics (See Table 1).

[1] We focused on UML because it is the most important OO modelling standard.

[2] Even though these metrics have been defined in a methodological way following a method proposed by Calero et al. [14], which consists of three main tasks: metric definition, theoretical and empirical validation, in this paper we focus only on empirical validation. Work related to their definition and theoretical validation can be found in Genero [13].

1530-1435/03 $17.00 © 2003 IEEE

Table 1. Metrics for UML class diagram structural complexity

Type of Metrics	Metric definition
Size metrics	Number of Classes (NC). The total number of classes.
	Number of Attributes (NA). The total number of attributes.
	Number of Methods (NM). The total number of methods.
Structural complexity metrics	Number of Associations (NASSOC).The total number of associations.
	Number of Aggregations (NAGG). The total number of aggregation relationships (each "whole-part" pair in an aggregation relationship).
	Number of Dependencies (NDEP). The total number of dependency relationships.
	Number of Generalisations (NGEN). The total number of generalisation relationships (each "parent-child" pair in a generalisation relationship).
	Number of Generalisation hierarchies (NGENH). The total number of generalisation hierarchies.
	Number on Generalisation hierarchies (NAGGH). The total number of aggregation hierarchies (whole-part structures.)
	Maximum DIT (MAXDIT). It is the maximum DIT value obtained for each class of the class diagram. The DIT value for a class within a generalisation hierarchy is the longest path from the class to the root of the hierarchy.
	Maximum HAGG (MAXHAGG). It is the maximum HAGG value obtained for each class of the class diagram. The HAGG value for a class within an aggregation hierarchy is the longest path from the class to the leaves.

As the proposal of metrics is of no value if their practical use is not demonstrated empirically [15], [16], [17], [18], our main motivation has been to investigate, through experimentation, whether the metrics we proposed for UML class diagram structural complexity and size could be good predictors of class diagram maintainability. If this is corroborated by several empirical studies, we really will have obtained early indicators of class diagram maintainability. These indicators will allow OO software designers to make better decisions early in the OO software development life cycle, thus contributing to the development of better quality OO software.

This paper has three objectives:

1. To find a prediction model which relates the metrics shown in Table 1 with maintainability measures. This is done using the data obtained through a controlled experiment carried out with students within the University of Castilla-La Mancha in Spain.
2. To confirm the findings using the data obtained in a replication of the experiment undertaken with students at the Università degli Studi di Roma "Tor Vergata" in Italy.
3. Finally, to evaluate the predictive accuracy of the models obtained.

This paper starts with a description of the related work and a comparison with the objectives of ourwork. Following that, a description of the empirical data from which the results are drawn is presented. Section 4, provides the data analysis and interpretation, and finally the last section presents some concluding remarks and outlines directions for future research activities.

2. Related work

As some studies which have reviewed the state of the art of empirical studies related to OO measures reveal [6], [15], [19], [20] the dependent variables investigated are either fault proneness (probability of fault detection), the number of faults or changes in a class, the effort of various development activities, or expert opinion about psychological complexity of a class.

Related to maintainability as a dependent variable, this being the subject we are occupied with in the present study, several works exist [8], [21], [22], [23], that have proposed prediction models for maintenance tasks. But in most of these studies the measurement of the independent variables was performed from the source code and not from UML class diagrams, and for this reason the predictions have been made later in the development. Despite this fact, Briand and Wüst [5], [6] and Card et al. [3], among others, highlighted that the earlier the measurement is taken the better.

This work is part of a project we have been developing during the last three years with the aim of looking for early indicators of UML class diagram maintainability. Here, we will briefly summarise our previous work:

- First, we thoroughly analysed the existing measures that could be applied to class diagrams at a high-level design stage, and proposed new ones which measure the structural complexity of class diagrams, due to the usage of UML relationships (See Table 1) [11], [12], [13].
- In [24] we presented an experiment where the subjects were given 24 class diagrams and they had to subjectively evaluate some maintainability sub-characteristics. Even though the opinion of the subjects is in nature subjective, the preliminary findings were encouraging. All of the metrics we proposed seemed to be related to maintainability sub-characteristics.
- In [25] we presented an experiment and its replica where the subjects were given nine class diagrams and they had to modify them to achieve some new requirements. In this case, we found that metrics related to aggregation and generalisation relationships are highly correlated to modifiability correctness and completeness.
- In [26] we described an experiment where the subjects were given nine class diagrams and had to modify them according to three new requirements, and had to note the time used in those modifications. As a result, we found that the maintenance time seemed to be correlated with all the metrics except those related to the number of dependencies.

3. Data description

The data used in this research was obtained through a controlled experiment carried out by students of the Department of Computer Science at the University of Castilla-La Mancha, in Spain[3] (See Section 3.1), and its replication carried out by students of the Dipartimento di Informatica, Sistemi e Produzione at the Università degli Studi di Roma "Tor Vergata", in Italy (See Section 3.2)

We have followed some suggestions provided by Wohlin et al. [27], Perry et al. [28], Briand et al. [4] and Kitchenham et al. [29] on how to perform controlled experiments. Due to space constraints, we outline only the main characteristics of the experimental process.

Some differences can be detected between the experiments that we conducted in Spain and Italy, respectively. For instance, a minor difference was that 24 subjects participated in Spain and 26 subjects in Italy. Because of some major differences, the doubt could be raised as to whether the latter can be considered as a replica of the former or should be regarded as a new experiment. In order to highlight this aspect, we will first enumerate similarities and differences between the two experiments,

[3] The experimental material can be found in http://alarcos.inf-cr.uclm.es

3.1. Similarities between the controlled experiment and its replication

- The subjects were advanced students of Computer Science. At the time of the experiment, all of the students had taken two courses in Software Engineering, in which they had studied in depth how to design OO software using UML. Moreover, subjects were given an intensive training session before the experiment took place.
- 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 ordered differently for each subject.
- The material we gave to subjects consisted of nine UML class diagrams of different application domains. Each diagram had an enclosed test that included a brief description of what the diagram represented, and two types of tasks:
  - Understandability tasks: where the subjects had to answer a questionnaire (4 questions) that reflected whether or not they had understood each diagram. In order to obtain the Understandability Time, expressed in minutes and seconds, subjects also had to note how long it took to answer the questions.
  - Modifiability tasks: Where the subjects had to modify the class diagrams according to four new requirements, and specify both the start and end time. The difference between the two times is what we call Modifiability Time (expressed in minutes and seconds). The modifications to each class diagram were similar, including adding attributes, methods, classes, etc.
- The subjects were given all the materials described in the previous point and we explained to them how to carry out the tests. The material was printed on paper supports.
- Each subject had to carry out the test alone.
- The independent variables are the structural complexity and the size of UML class diagrams, measured using the metrics shown in Table 1.
- The dependent variables are two maintainability sub-characteristics modifiability and understandability, measured as is usual in empirical studies [23], [30]:
  - Understandability Time is time spent by the subjects answering the understandability questions.
  - Modifiability Time is the time spent by the subjects doing the modifications tasks.
    - Modifiability Correctness =

$$\frac{\textit{Number Of Correct Modifications}}{\textit{Number Of Modifications Applied}}$$

- Modifiability Completeness =

$$\frac{Number\ Of\ Correct\ Modifications}{Number\ Of\ Modifications\ Required}$$

Table 2. Metric values of the UML class diagrams used in the experiment

Diagram	NC	NA	NM	NAssoc	NAgg	NDep	NGen	NAggH	NGenH	MaxHAgg	MaxDIT
1	13	30	47	11	5	3	3	2	1	2	1
2	22	43	56	11	6	4	15	1	7	4	3
3	9	24	25	6	1	1	5	1	2	1	2
4	7	17	26	1	4	0	3	2	1	1	1
5	9	24	40	2	3	1	5	1	2	1	1
6	11	26	36	5	0	2	7	0	4	0	3
7	52	76	35	15	23	8	17	3	6	7	4
8	22	62	39	7	12	0	2	4	1	1	1
9	7	23	31	3	1	1	2	1	1	1	1

3.2. Basic experiment

The experiment that we conducted in Spain can also be characterized as the following:

- Subjects were 24 undergraduate students enrolled on the third-year of Computer Science at the Department of Computer Science at the University of Castilla-La Mancha in Spain.
- Class diagrams and other experiment material were written in Spanish.
- We allowed subjects one week to undertake the experiment, i.e. each subject had an unlimited amount of time to solve the test.

3.3. Replication of the experiment

The experiment that we conducted in Italy does not vary the hypotheses of the basic experiment [15], [31]. We hence see it as a replication; however, exceptions can be raised against the use of such a word and concept, and make any generalization questionable. We should also note that two arrangements for conduction of the experiment were available. One expectation was to use paper supports, with experiment material translated from Spanish to English by the Spanish team. Another expectation was to conduct the experiment through the Internet with experiment material translated from English to Italian by the Italian team. We eventually decided to use the former, but give subjects authorization to access the Italian translation on line.

The experiment that we conducted in Italy can also be characterized as the following:

- The subjects were 26 undergraduate students enrolled on the fourth year of Computer Science at the Dipartimento di Informatica, Sistemi e Produzione at the Università degli Studi di Roma "Tor Vergata", in Italy.
- The experiment material that the subjects had to work on was translated in English (See Appendix A for an example), with on-line help in Italian. This might have biased the results because on-line help in the native language was not always semantically clear, not all the subjects have enough knowledge of the English language, and they occasionally needed extra time to ask the professor who monitored the experiment about the meaning of some statements.
- We allowed subjects two hours to undertake the experiment; i.e. after training anddebriefing, each subject had 2 hours to solve the test. In fact, the Italian experiment was carried out in a more controlled environment due to the fact that it was supervised.

4. Data analysis and interpretation

The main goal of these experiments was to analyze class diagrams for the purpose of investigating the possibility of the structural complexity and size metrics of class diagrams being used as good predictors of class diagram maintainability, from the point of view of the researcher in the context of students from an Italian an a Spanish University. Therefore, we propose the following hypotheses:

- H_0: the structural complexity and size metrics of class diagrams can be used as good predictors of class diagram maintainability.
- H_1: the structural and size metrics of class diagrams cannot be used as good predictors of class diagram maintainability.

As we have proposed several measures for class diagram maintainability (dependent variables), such as Understandability Time, Modifiability Time, Modifiability Correctness, Modifiability Completeness (See Section 3.1), these hypotheses were specified for each dependent variable in three data sets:

- The data obtained from the controlled experiment described in Section 3.1, which we have called "Spain data".

- The data obtained through the replication of the experiment, described in Section 3.2, which we have called "Italy data".
- Both data sets together, which we have called "All data".

In order to test our hypotheses we have selected the following Multivariate Lineal Model:

$$Y = \mu + \sum_{j=1}^{r} \beta_j X_j + \varepsilon$$

Where Y is one of the dependent variables, and X_j are the independent variables that explain Y significantly, ε are $N(0, \sigma)$, and μ is the intercept . However, this does not always have an easy interpretation. Above all, we are interested in β_j (partial correlation coefficient), and this means that if X_j is incremented in one, Y is incremented in β_j.

In software engineering experimentation (Briand and Wüst, 2002; Mendes et al., 2002), multivariate analysis is commonly used because it takes into account the relationships between independent and dependent variables. However, it also considers the former by way of combination as covariates in a multivariate model in order to explain the variance of the dependent variables in a better way, and ultimately obtain accurate predictions.

The selection of the independent variables can be carried out using different methods for example stepwise backward or forward selection. The general forward selection procedure starts with a model, which only includes the intercept, and the independent variables are selected one at a time for inclusion in the model, as long as they fulfil certain statistical criteria. Similarly, the backward procedure starts with a model which includes all the independent variables, which are selected one at a time to be deleted from the model if comply with certain statistical criteria. Both procedures stop when a criterion is fulfilled (we have used 0.05 probability for inclusion and 0.10 for exclusion)[4].

As the selected model requires data to be independent, we have randomly considered k different subjects for each diagram to assure data independence (See Table 3)[5].

We carried out the following steps in order to analyse the empirical data [6]:

- First, we have analysed the descriptive statistics for the dependent variables (See Section 4.1).
- Then, we have modelled the relationship between the independent variables and each dependent variable using a multivariate lineal model (See Section 4.2).
- After that, we have validated the models to see whether the residuals complied with the hypotheses of the model or not (normality, independence etc.) (See Section 4.3).
- Finally, we have evaluated the predictive accuracy of the models (See Section 4.4).

Table 3. Assignation of diagrams to subjects

S→ subjects from Spain				I→ subjects from Italy					
Diagram 1		**Diagram 2**		**Diagram 3**		**Diagram 4**		**Diagram 5**	
S1	S4	S5	S8	S3	S6	S2	S7	S9	S19
S17		S22		S24		S23		I9	I19
I1	I4	I5	I8	I3	I6	I2	I7	I25	
I17		I22		I24		I23			
Diagram 6		**Diagram 7**		**Diagram 8**		**Diagram 9**			
S11	S16	S13	S18	S10	S14	S12	S15		
I11	I16	I13	I18	S20		S21			
		I26		I10	I14	I12	I15		
				I20		I21			

4.1. Descriptive Statistics

Descriptive statistics of the dependent variables are presented in Table 4, Mean, Standard Error (SE), Median and Inter-Quartile Range (IQR). The descriptive statistics for Understandability Time and Modifiability Time have higher values in Italy data compared with Spain data, which may have occurred because, as we said in Section 3.2, the subjects from Italy did not use their native language (the experiment was in English), so they required a certain amount of necessary extra time to undertake the required tasks.

Table 4. Descriptive statistics for dependent variables

	Italy (n=26)			
Dependent variables	**Mean**	**SE**	**Median**	**IQR**
Understandability Time	209.880	27.740	166.000	194.500
Modifiability Time	511.190	33.270	497.500	288.250
Modifiability Correctness	0.880	0.039	1.000	0.180
Modifiability Completeness	0.760	0.049	0.850	0.420

[4] The theory underlying the multivariate analysis appears in a lot of statistical books such as Snedecor [33] or Kleinbaum [34], and its application to empirical studies is summarised by Briand and Wüst [6].

[5] We are aware that we are wasting data, but when we designed the experiment we had thought to analyze the data using univariate analysis, such as Spearman or Pearson coefficients instead of multivariate analysis. This must be considered in the next studies that we carry out.

[6] The 11.0 version of SPSS [35] has been used to extract the statistical data information.

	Spain (n=24)			
Dependent variables	**Mean**	**SE**	**Median**	**IQR**
Understandability Time	92.960	11.790	85.000	77.250
Modifiability Time	265.880	26.620	261.000	168.750
Modifiability Correctness	0.850	0.042	0.880	0.250
Modifiability Completeness	0.810	0.044	0.810	0.310

4.2. Model Selection

First of all we took into account the Principal Component Analysis (PCA) carried out in order to select the independent variables with high loadings in the rotated components (These were NA, NAGG, NDEP, NGENH and MAXDIT) [36]. Afterwards, we estimated models for each dependent variable following the stepwise backward and forward selection of the independent variable. The problem with the forward selection in the selected models was collinearity. Only the Understandability Time and Modifiability Completeness in the Italy data did not have collinearity problems, so finally we used the backward procedure. From this point of view, the criterion used to select the models was simpler, with less p-value of F-test and the better goodness of fit (R^2).

The models selected are in Tables 5 to 7. These tables have:

- The ANOVA results which permit contrast of the hypothesis:
  - H_0: the independent variables in the lineal model not explained by the dependent variable
  - H_1: $\neg H_0$
  - SS-model→ Squared Sum of variance explained by the model
  - SS-residual → Squared Sum of variance not explained by the model
  - d.f.→ degree of freedom of SS
  - p-value→ if p-value is less than 0.05 the model is accepted
  - The adjusted lineal model
- R^2 → Goodness of fit, which refers to the percentage of the variability of the dependent variable explained by the model.

Table 5. Understandability Time models

		p-value	SS-model d.f.	SS-residual d.f.	R^2
Spain data	25.784 + 0.522 * NA	0.001	28788.548 1	478921.411 22	0.375
Italy data	73.676 + 73.779 * NAGGH	0.002	170839.066 1	329295.588 24	0.342
All data	59.772 + 52.802* NAGGH	0.000	174477.389 1	572961.731 48	0.233

Table 6. Modifiability Correctness models

		p-value	SS-model d.f.	SS-residual d.f.	R^2
Spain data	1.402 – 0.114 NAGGH	0.001	0.399 1	0.581 22	0.408
Italy data	0.992 – 0.040 NGENH	0.020	0.206 1	0.800 24	0.205
All data	1.005 – 0.078 NAGGH	0.002	0.380 1	1.622 48	0.190

Table 7. Modifiability Completeness models

		p-value	SS-model d.f	SS-residual d.f.	R^2
Spain data	1.004 – 0.116 * NAGGH	0.001	0.420 1	0.628 22	0.401
Italy data	1.079– 0.009 * NA	0.000	0.763 1	0.812 24	0.485
All data	1.072 –0.008 NA	0.000	1.172 1	1.483 48	0.441

Next, we will analyse the content of Tables 5, 6 and 7:

- Looking at the Italy data in Table 5, we can see that the model explains Understandability Time, with NAGGH as an explanatory variable. Each unit incremented in NAGGH produces a 73.779 seconds increase in the Understandability Time, and 73.676 seconds is the intercept (μ estimation). The p-value is 0.001, significant at 0.05 level. It means that we can accept this model to explain Understandability Time in the Italy data, the smaller the p-value the better the model. The value of R^2 means that NAGGH explains 34.5% of Understandability Time variation in the Italy data. Similarly, we can interpret the results of Table 5 for the Spain data and for the All data.
- Table 6 shows Modifiability Correctness. If we look at the Spain data, we can see that each unit incremented in NAGGH produces a reduction of –0.114 in Modifiability Correctness. We can accept this model because the p-value is 0.001, and the model explains 40.8% of Modifiability Correctness variation in the Spain data. Similarly, we can interpret the other models in Table 6.
- Finally Table 7 has the Modifiability Completeness models. Again, in the Spain data NAGGH significantly explains the Modifiability Completeness. We can see that each unit incremented in NAGGH produces a reduction of –0.116 in Modifiability Completeness. We can accept this model because the p-value is 0.001 and the model explains 40.1% of Modifiability Completeness variation in the Spain data. Similarly, we can interpret the other models in Table 7.

In summary, Tables 5 to 7 show that measures of aggregation and generalization relationships, and the number of attributes are determinant for class diagram maintainability because they are the only explanatory variables in these models. The models change if all data is considered as a whole rather than separately. Even though the p-values are similar, the goodness of fit is inferior when compared with that of the Italy data and the Spain data. Only in the Modifiability Completeness model (Table 7) is the goodness of fit of the Spain data superior, and the Italy one is inferior.

The Modifiability Time Table was omitted because neither models nor their transformations in Modifiability Time (Ln, sin, etc.), nor other non-lineal models adjust significantly. However, the Modifiability Correctness and Completeness, as we can see, have models that adjust significantly. Furthermore, it is possible to confirm the external validation of these results in other experiments, using class diagrams with different independent variable values and different subjects.

4.3. Model Validation

One of the threats to conclusion validity is the violating assumptions of statistical tests. That is why we have studied a lineal model validation of the homogeneity of variance, normal distribution and independence of residuals [34].

- Normality of residuals
  - H_0: the standardised residual has normal distribution
  - H_1: $\neg H_0$

  We have tested these hypotheses with the Kolmogorov-Smirnoff and the Shapiro-Wilk test. As we can see looking at the p-values of Table 8, the conclusion is that, at 0.05 level, there is a lack of normality in the All data andthe Italy data when Modifiability Correctness and Modifiability Completeness are considered. Nevertheless, the selected model is robust with respect to these hypotheses, thus we can accept conclusions based on this model.
- Independence of residuals

  The Durbin-Watson test contrasts the following hypotheses [37]:
  - H_0: the residuals do not have first-order autocorrelation
  - H_1: $\neg H_0$

  Table 9 shows that the results were significant when $\alpha = 0.05$ except with

regards to the Spain data with Modifiability Correctness and Modifiability Completeness. In this case, the tests are inconclusive at the 0.05 level, but significant at 0.01 level. This means that we do not have a solid argument against the H_0.

Table 8. Residual Normality Test

	Understandability Time		Modifiability Correctness		Modifiability Completeness	
	Kolmogorov-Smirnoff	Shapiro-Wilk	Kolmogorov-Smirnoff	Shapiro-Wilk	Kolmogorov-Smirnoff	Shapiro-Wilk
Spain data	0.20	0.583	0.20	0.389	0.098	0.137
Italy data	0.20	0.287	0.01	0.07	0.072	0.003
All data	0.01	0.010	0.01	0.01	0.001	0.001

Table 9. Durbin Watson Test

	Understandability Time	Modifiability Correctness	Modifiability Completeness
Spain data	1.650	1.433	1.237
Italy data	1.971	2.349	2.228
All data	1.932	1.790	1.959

- Homogeneity of variance
  To explore the homocedasticity (homogeneity of variance), we looked at the scatter diagrams of standardised residuals against prognosticated standardised values, and they did not show any deviation in form. Furthermore, the selected models appear to be valid because these graphics did not show any regular shape.

Finally, we have searched for the influential points in the models. The SPSS provides some statistics which can detect the influential points, such as Cook's distance, adjusted difference (DFFIT), etc. [34]. We have made an exploratory analysis using these statistics, and no influential points have been found. To illustrate this, figures 1 and 2 show these two distances in the adjusted model for Understandability Time in the Spain data.

4.4. Models predictive accuracy

We have used the mean magnitude of relative error (MMRE), quartiles of MRE distribution and Pred (n) to evaluate the predictive accuracy of the models [6], [31], [38], [39].

As Tables 10 to 12 show, the Understandability Time lineal model seems to be a better predictive model in the Spain data. The Modifiability Completeness and Modifiability Correctness models seem to be very good predictive models. According to the models, we found that if we know the number of aggregation hierarchies (NAGGH), attributes (NA) and generalization hierarchies (NGENH) of a class diagram, we can accurately predict its Understandability Time, its Modifiability Completeness and its Modifiability Correctness for the maintenance process.

Table 10. Prediction accuracy of the Understandability Time model

	MMRE	P_{25}	P_{50}	P_{75}	PRED (0.30)
Spain data	0.583	0.193	0.313	0.630	46.0 %
Italy data	0.676	0.192	0.376	0.680	38.5 %
All data	0.850	0.240	0.462	0.876	30.0 %

Table 11. Prediction accuracy of the Modifiability Completeness model

	MMRE	P_{25}	P_{50}	P_{75}	PRED (0.30)
Spain data	0.242	0.086	0.143	0.228	87.5 %
Italy data	0.3675	0.061	0.129	0.220	84.6 %
All data	0.321	0.080	0.38	0197	84.0 %

Table 12. Prediction accuracy of the Modifiability Correctness model

	MMRE	P_{25}	P_{50}	P_{75}	PRED (0.30)
Spain data	0.248	0.060	0.079	0.190	95.8 %
Italy data	0.257	0.048	0.088	0.192	88.5 %
All data	0.282	0.059	0.115	0.189	88.0 %

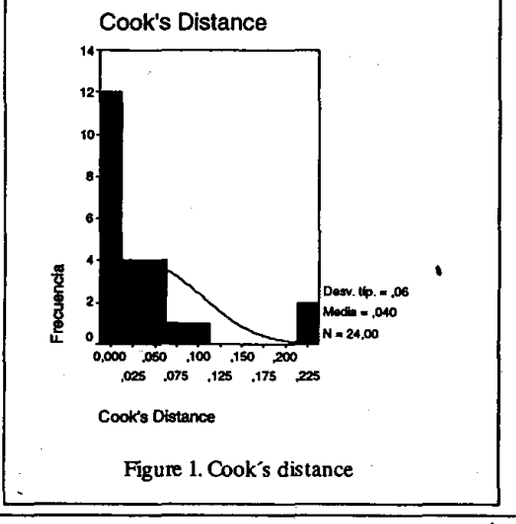


Figure 1. Cook's distance

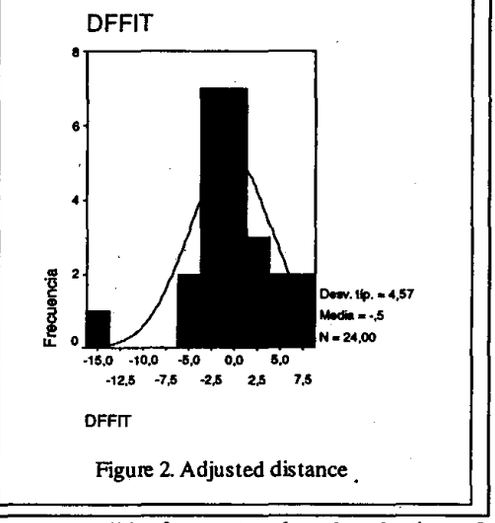


Figure 2. Adjusted distance

Tables 10 to 12 show that predictive values are inferior when the data is considered as a whole, which confirms the conclusions of Section 4.2. Therefore, we think that the Italy and Spain data must be studied separatelybecause it seems that there is a place effect (students with different backgrounds, different language used in the material, etc.)."

As we said in Section 4.2, we had found other models using forward selection for Understandability Time and Modifiability Completeness in the Italy data. They confirm the correlation between NAGGH and NGENH with the dependent variables; otherwise, it is true that other independent variables appear in the models as HAGGMAX, NGEN and MAXDIT, which are also related to aggregation and generalisation relationships.

4.5. Threats to validity

We will discuss, in turn, threats to the conclusion, construct, internal, and external validity. Our goal here is firstly to help the readers qualify the results that are presented in this paper, and secondly, propose future research by highlighting some of the issues associated with our study.

4.5.1. Conclusion validity. Conclusion validity is the degree to which conclusions can be drawn about the existence of a statistical relationship between treatments and outcomes. In both experiments, due to participation on voluntary basis and because of the small population, it was no possible for us to plan the slection of a population sample by using one of the common sampling techniques, so we decided to take the whole population of the available classes as our target samples. A limited nimber of data values were collected during the excecution of the experiments, due to the limited duration time and the number of subjects. For what concerns the quality of data collecting, we used pencil and paper; hence data colection could consider being critical. Finally the quantity and the quality of the data collected and the data analysis were enough to support our conclusions, as described in previous sections, concerning the existence of a statistical relationship between the independent and dependent variables, i.e. some of the metrics investigated have a statistical relationship with Understandability Time, Modifiability Correctness and Modifiability Correctness.

4.5.2. Construct validity. Construct validity is the degree to which the independent variables and dependent variables accurately measure the concepts they purport to measure.

The dependent variable: class diagram maintainability, was measured by using different times. These are Understandability and Modifiability Times, which are objective measures that reflect the time that the subjects spent solving the experimental tasks. For this reason we consider that they objectively measure what we purport to measure. We also consider the Modifiability Correctness and Completeness, which to some extent reflect how well the subjects undertake the modifiability tasks.

To investigate the conctruct validity of the structural complexity and size measures used in this study, the reader could refer to [13].

4.5.3. Internal validity. Internal validity is the degree to which conclusions can be drawn about the causal effect of the independent variables. The analysis performed here is correlational in nature. We have demonstrated that some of the metrics investigated have a statistically and practically significant relationship with Understandability Time, Modifiability Correctness and Modifiability Completeness. Such statistical relationships do not intrinsically demonstrate a causal relationship. They only provide supporting evidence of it. Only controlled experiments, where the measures (the independent variables studies in the experiment) are varied in a controlled manner and all other independent variables are held constant, could really demonstrate causality. However, such a controlled experiment would be difficult to run since varying size and structural complexity in a system, while preserving its functionality, is difficult in practice.

4.5.4. External validity. External validity is the degree to which the results of the research can be generalised to the population under study and other research settings. The greater the external validity, the more the results of an empirical study can be generalised with regards to actual software engineering practice. Two threats to validity have been identified which limit the ability to apply any such generalisation:

- Materials and tasks used. In the experiment, we tried to use class diagrams and tasks representative of real cases, but more empirical studies, using "real cases" from software companies, must be carried out.
- Subjects. To solve the difficulty of obtaining professional subjects, we used students from advanced software engineering courses. We are aware that more experiments with professionals must be carried out in order to be able to generalise these results. However, in this case, the tasks to be performed do not require high levels of industrial experience, therefore, experiments with students may well be appropriate [15][7].

5. Conclusions

It is widely recognized that the quality of OO software must be assessed early, based on software analysis and design artefacts. This fact lead us to define a set of metrics for assessing the structural complexity and size of UML class diagrams, with the hypothesis being that they are correlated with the maintainability of such diagrams. To corroborate this, in this study we have presented prediction models for different class diagram maintainability measures, such as Understandability Time, Modifiability time, Modifiability Correctness and Modifiability Completeness. These models are based on the 11 early metrics we proposed (See Table 1), which measure class diagrams structural complexity and size. The data was taken from one experiment carried out by students in Spain and its replication performed by students in Italy.

After a multivariate analysis, we can conclude that the obtained multivariate lineal models have proved the Understandability Time, the Modifiability Correctness and Modifiability Completeness to be related to the structural complexity measures NAGGH (number of aggregation hierarchies) and NGENH (number of generalisations hierarchies), and with some of the size measures, NA (number of attributes).

The obtained Modifiability Time models do not significantly adjust. However, at this stage we cannot affirm that those metrics are not good predictors of Modifiability Time, without performing more empirical studies.

The problems with the hypotheses of the multivariate lineal models might be due to the sample size being small. Nevertheless, predictive accuracy of the models has been effective, especially regarding Modifiability Completeness and Modifiability Correctness.

Based on the information already presented, we can safely say that metrics concerning aggregation and generalisation relationships are highly related to the Understandability Time and Modifiability Correctness and Completeness. From a practical point of view, this means that when class diagram modification tasks are required, considering the values of aggregation and generalisation metrics, and also the number of attributes, it may be useful to predict the level of correctness and completeness of those tasks. It may also be helpful to predict the time subjects need to understand a class diagram before modifying it.

These findings could be very valuable when predicting the maintenance of OO software products, one of the biggest concerns in software organisations.

From the results presented in this study, we may conclude that there is a reasonable chance that useful class diagram maintainability models could be built at the initial phases of the OO software life cycle (e.g. when choosing between two semantically equivalent design alternatives), thus allowing OO software designers to make better decisions early in the OO software development life cycle. Nevertheless, we do not believe that universally valid quality measures and models can be devised at this stage. Therefore, the focus of our multivariate analysis is to obtain an initial assessment of the feasibility of building class diagram maintainability prediction models based on early

[7] Nevertheless, we are aware that in laboratory experiments carried out by students, such as our experiment, the objects are stand-alone, i.e. the class diagrams does not have a context. This fact motivates us even more to carry out experiments using real projects in industrial settings.

metrics. However, as Briand and Wüst [5] remarked, early analysis and design artefacts are, by definition, not complete, and only represent early models of the actual system to be developed. For this reason, the use of predictive models based on early artefacts, and their capability to predict the quality of the final system still remains to be investigated. Further replication is of course necessary to build an adequate body of knowledge regarding the use of OO early measures. Moreover, the study performed in this paper should be replicated in a variety of environments and systems in order for our community to draw general conclusions about what OO measures can do to help assess the quality of early designs and systems.

As Miller [40] and Basili et al. [15], among others, suggested, simple studies rarely provide definite answers. Following these suggestions, we have carried out a family of experiments, including the experiment presented in this paper. We are aware that only after performing a family of experiments can you build an adequate body of knowledge to extract useful measurement conclusions regarding the use of OO design metrics to be applied to real measurement projects [2], [15].

Some changes that could be made to improve this experiment are:

- Increasing the size of the class diagrams. By increasing the size of the class diagrams, we have examples that are closer to reality. In addition, if we are working with professionals, we can make better use of their potential capability and conclude that the results are more general.
- Increasing the difference between the values of the metrics. This option could lead to results which are more conclusive about the metrics and their relationship with the factor we are trying to control.
- Improving the design of the experiment in order not to waste empirical data (See Section 3.1.), dividing the subjects in groups, and each group given different class diagrams.
- Carrying out the experiment in a more controlled environment, using a modeling tool such as Rational Rose, and not the class diagrams written on paper.
- Work with real data obtained from industrial environments. However, the scarcity of such data continues to be a great problem so we must find other ways of tackling the validation of metrics.
- As Brito e Abreu et al. [41], [42], [43] and Basili et al. [15] suggested it is necessary to have a public repository of laboratory packages, which we think would be a good step towards the success of all the work carried out on software measurement.

Acknowledgements

This research is both part of the DOLMEN (TIC 2000-1673-C06-06) and CALDEA (TIC 2000-1673-C06-06) projects, financed by Subdirección General de Proyectos de Investigación - Ministerio de Ciencia y Tecnología, Spain, and the MIUR IUC 2001 "Experimental Informatics in Europe" Inter University Cooperation project, financed by Ministero dell'Istruzione, dell'Università e della Ricerca, Italy, Grant No. URM2 DISP 020906003.

Appendix A

Here we provide as an example Diagram 4 and its related tasks. For better understanding, we present the version given to subjects in Italy because it is in English.

Diagram 4

With the UML class diagram shown below, you have to perform the following tasks:

Tasks: Part I

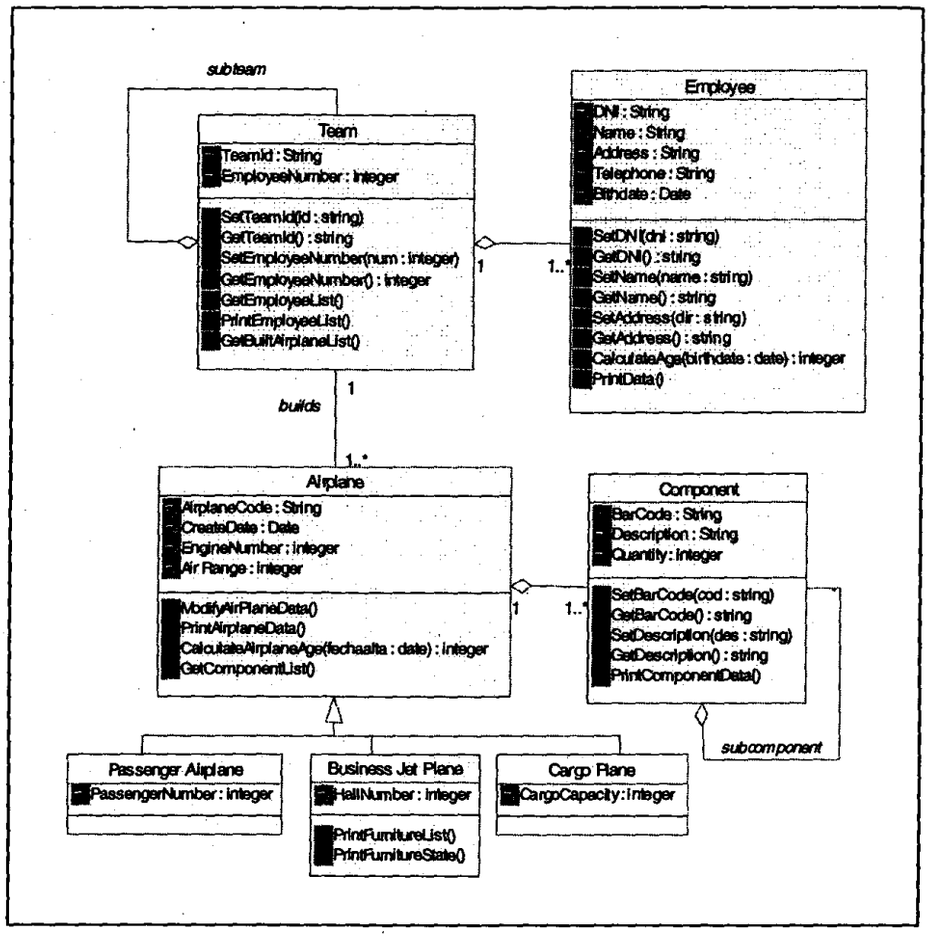
*** Answer the following questions:**
Write down the starting hour (indicating hh:mm:ss):

_____

1.- Can a component be composed of other components? _____
2.- Can an air plane be built by several teams? _____
3.- Can an employee belong to different teams? _____
4.- Can a team build several airplanes?_____
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.- We want to know the date when an employee starts to work in a team.
2.- We want to obtain all the information both in printer and screen of a work team.
3.- We want to specify that the employees can be temporary or permanent. To the temporary employees we want to know the date when their contract of employment finishes.
4.- We want to specify that an employee belong only to one work team and a work team is composed of two or more employees.
Write down the ending hour (indicating hh:mm:ss):



References

[1] N. Schneidewind, N., "Body of Knowledge for Software Quality Measurement", *IEEE Computer*, 35(2), 2002, pp. 77-83.

[2] L. Briand, J. Wüst and H. Lounis, "Investigating Quality Factors in Object-oriented Designs: An Industrial Case Study", *Technical Report ISERN 98-29 (version 2)*, 1998.

[3] D. Card, K. El-Emam and B. Scalzo, "Measurement of Object-Oriented Software Development Projects", Software Productivity Consortium NFP, 2001.

[4] L. Briand, S. Arisholm, F. Counsell, F. Houdek and P. Thévenod-Fosse, "Empirical Studies of Object-Oriented Artefacts, Methods, and Processes: State of the Art and Future Directions", *Empirical Software Engineering*, 4(4), 2000, pp. 387-404.

[5] L. Briand and J. Wüst, "Modeling Development Effort in Object-Oriented Systems Using Design Properties.", *IEEE Transactions on Software Engineering*, 27(11), 2001, pp. 963-986.

[6] L. Briand and J. Wüst, "Empirical Studies of Quality Models in Object-Oriented Systems", *Advances in Computers*, Academic Press, Zelkowitz (ed.), 59, 2002, pp. 97-166.

[7] J. Bansiya and C. Davis,"A Hierarchical Model for Object-Oriented Design Quality Assessment", *IEEE Transactions on Software Engineering*, 28(1), 2002, pp. 4-17.

[8] F. Fioravanti and P. Nesi, "Estimation and Prediction Metrics for Adaptive Maintenance Effort of Object-Oriented Systems", *IEEE Transactions on Software Engineering*, 27(12), 2001, pp. 1062-1083.

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

[10] Pigoski, T., "Practical Software Maintenance", Wiley Computer Publishing, New York, USA, 1997.

[11] M. Genero, Piattini M. and C. Calero, "Early Measures For UML class diagrams", L´Objet, 6(4), Hermes Science Publications, 2000, pp. 489-515.

[12] M. Genero, Piattini M., and C. Calero, "Metrics for high-level design UML class diagrams: an exploratory analysis. Submitted to *Software Quality Journal*, 2003.

[13] M. Genero, "Defining and Validating Metrics for Conceptual Models", *Ph.D. thesis*, University of Castilla-La Mancha, 2002.

[14] C. Calero, Piattini M. and M. Genero, "Empirical validation of referential integrity metrics", *Information and Software Technology*, 43, 2001, pp. 949-957.

[15] V. Basili, Shull F. and Lanubile F., "Building Knowledge through Families of Experiments", *IEEE*

Transactions on Software Engineering, 25(4), 1999, pp. 435-437.

[16] B. Kitchenham, Pfleeger S. and Fenton N., "Towards a Framework for Software Measurement Validation", *IEEE Transactions of Software Engineering*, 21(12), 1995, pp. 929-943.

[17] N. Schneidewind, "Methodology For Validating Software Metrics", *IEEE Transactions of Software Engineering*, 18(5), (1992), 410-422.

[18] G. Cantone and P. Donzelli, "Production and maintenance of software measurement models", *Journal of Software Engineering and Knowledge Engineering*, 5, 2000, pp. 605-626.

[[19] K. El-Emam, "Object-Oriented Metrics: A Review on Theory and Practice", *NRC/ERB 1085*, National Research Council Canada, 2001.

[20] I. Deligiannis, M. Shepperd, S. Webster and M. Roumeliotis, "A Review of Experimental into Investigations into Object-Oriented Technology", *Empirical Software Engineering*, 7(3), 2002, pp. 193-231

[21] W. Li and S. Henry, "Object-Oriented Metrics that Predict Maintainability", *Journal of Systems and Software*, 23(2), 1993, pp. 111-122.

[22] R. Harrison, S. Counsell and R. Nithi, "Experimental Assessment of the Effect of Inheritance on the Maintainability of Object-Oriented Systems", *Journal of Systems and Software*, 52, 2000, pp. 173-179.

[23] L. Briand, C. Bunse and J. Daly, "A Controlled Experiment for evaluating Quality Guidelines on the Maintainability of Object-Oriented Designs", *IEEE Transactions on Software Engineering*, 27(6), 2001, pp. 513-530.

[24] M. Genero, J. Olivas, M. Piattini and F. Romero, "Using metrics to predict OO information systems maintainability", *CAISE 2001*, Lecture Notes in Computer Science, 2068, Interlaken, Switzerland, 2001, pp. 388-401.

[25] M. Genero, L. Jiménez L. and M. Piattini, "A Controlled Experiment for Validating Class Diagram Structural Complexity Metrics". *The 8th International Conference on Object-Oriented Information Systems (OOIS'2002)*, Lecture Notes in Computer Science 2425. Bellahsene, Z., Patel, D., Rolland, C., (Eds.). Springer-Verlag, 2002, pp. 372-383.

[26] M. Genero, J. Olivas, M. Piattini and F. Romero, "Assessing object-oriented conceptual models maintainability", *International Workshop on Conceptual Modeling Quality (IWCMQ 2002)*. Lecture Notes in Computer Science, Springer, 2002 (to appear).

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

[28] D. Perry, A. Porter and L. Votta, "Empirical Studies of Software Engineering: A Roadmap", *Future of Software Engineering, ACM*, (Ed. Anthony Finkelstein), 2000, pp. 345-355.

[29] B. Kitchenham, S. Pfleeger, L. Pickard, P. Jones, D. Hoaglin D., K. El-Emam and J. Rosenberg, "Preliminary Guidelines for Empirical Research in Software Engineering", *IEEE Transactions on Software Engineering*, 28(8), 2002, pp.721-734.

[30] G. Poels and G. Dedene, "Evaluating the Effect of Inheritance on the Modifiability of Object-Oriented Business Domain Models", *5th European Conference on Software Maintenance and Reengineering (CSMR 2001)*, Lisbon, Portugal, 2001., pp. 20-29.

[31] E. Mendes, I. Watson, N. Mosley and S. Counsell, " A Comparison of Development Effort Estimation Techniques for Web Hypermedia Applications" *Eighth IEEE Symposium on Software Metrics (METRICS'02)*, 2002, pp. 21-30.

[32] A. Brooks, J. Daly, J. Miller, M. Roper and M. Wood, "Replication of experimental results in software engineering", *Technical Report ISERN-96-10*, International Software Engineering Research Network, 1996.

[33] Snedecor, G., and W. Cochran, *Statistical Method,s* 8 ª ed,. Iowa State University Press, 1989.

[34] Kleinbaum, D., L. Kupper, K., and Muller K., *Applied regression analysis and other multivariate methods*, second ed., Duxbury Press, 1987.

[35] SPSS 11.0., *Syntax Reference Guide*,. Chicago, SPSS Inc., 2001.

[36] Mª Manso, M. Genero and M. Piattini, "No-Redundant Metrics for UML Class Diagrams Structural Complexity". *CAISE 2003*, (to appear).

[37] Bovas, A., and J. Ledolfer, *Statistical Methods for Forecasting*, Wiley Series in Probability and Mathematical Statistics, 1983.

[38] L. Briand and I. Wieczorek, "Resource Estimation in Software Engineering", *Encyclopedia of Software Engineering*, second edition, Wiley, 2001.

[39] M. Shepeprd, C. Schofield and B. Kitchenham,. "Effort estimation using analogy", *18th ICSE*, IEEE Computer Society Press, 1996, pp. 170-178.

[40] J. Miller, "Applying Meta-Analytical Procedures to Software Engineering Experiments", *Journal of Systems and Software*, 54, 2000, pp. 29-39.

[41] F. Brito e Abreu, H. Zuse, H. Sahraoui and W. Melo, *"Quantitative Approaches in Object-Oriented Software Engineering. ECOOP'99 Workshops Reade!*, LNCS 1743, Springer-Verlag,1999, pp. 326-337.

[42] F. Brito e Abreu, G. Poels, H. Sahraoui and H. Zuse, " *Quantitative Approaches in Object-Oriented Software Engineering. ECOOP'2000 Workshop Reade"r*, LNCS 1964, Springer-Verlag, 2000, pp. 93-103.

[43] F. Brito e Abreu, B. Henderson-Sellers, M. Piattini, G. Poels and H. Sahraoui, *"Quantitative Approaches in Object-Oriented Software Engineering. ECOOP'01 Workshop Reader"*, LNCS 2323, Springer-Verlag, 2002, pp. 174-183.