Using Web Objects for Development Effort Estimation of Web Applications: A Replicated Study

Sergio Di Martino[1], Filomena Ferrucci[2], Carmine Gravino[2], and Federica Sarro[2]

[1] University of Napoli "Federico II", 80126, Napoli, Italy
sergio.dimartino@unina.it
[2] University of Salerno, Via Ponte Don Melillo, 84084, Fisciano (SA), Italy
{fferrucci,gravino,fsarro}@unisa.it

Abstract. The spreading of Web applications has motivated the definition of size measures suitable for such kind of software systems. Among the proposals existing in the literature, Web Objects were conceived by Reifer specifically for Web applications as an extension of Function Points. In this paper we report on an empirical analysis we performed exploiting 25 Web applications developed by an Italian software company. The results confirm the ones obtained in a previous study and extend them in several aspects, showing the robustness of the measure with respect to the size and technologies of the applications, and to the employed estimation techniques.

Keywords: Web applications, Size measure, Effort estimation technique, Empirical studies.

1 Introduction

Even if Web applications are becoming the de-facto standard in many domains, such as B2B [7], software engineering is still missing to fully support their development. Among others, there is to date the need of suitable measures to size this kind of applications and support critical management activities, such as cost/effort estimation, quality control, and productivity assessment. Indeed, size measurement methods, conceived and widely accepted for traditional software systems, such as the Function Point Analysis (FPA) [14], can fail to capture some specific features of Web applications [25]. Some measures have been defined so far (see e.g., [10][11][26]), and among them, *Web Objects* were introduced by Reifer [26] by adding four new Web-related components (namely Multimedia Files, Web Building Blocks, Scripts, and Links) to the five function types of the FPA method. In his original formulation [26][27], Reifer reported improved prediction performances of Web Objects over Function Points, but no details were provided about the empirical analysis he performed. In [29][30] Ruhe *et al.* described two studies assessing the effectiveness of Web Objects for estimating Web application development effort, by exploiting a dataset of 12 industrial Web applications. In the first analysis [30], they applied Ordinary Least Squares Regression (OLSR) [24], a widely used estimation technique, while in the second analysis [29] they employed Web-COBRA that is an extension for

D. Caivano et al. (Eds.): PROFES 2011, LNCS 6759, pp. 186–201, 2011.

Web applications of the COBRA[1] method proposed by Briand *et al.* [4]. Web-COBRA can be considered a composite method, according to a widely accepted taxonomy [3], since it exploits expert's opinions, gathered in a controlled fashion, together with other cost drivers, within an algorithmic approach. To assess the obtained estimations, authors applied a leave-1-out cross validation, highlighting that Web Objects performed well and better than Function Points. It is obvious that, as the authors themselves pointed out, there is the need of replicated studies to further assess the measure with different (and possibly) larger datasets and to generalize the results in different contexts [2]. To this aim, in this paper we report on a replication of Ruhe *et al.*'s studies [29][30] performed by exploiting data on 25 Web applications developed by an Italian software company. This analysis also extends Ruhe *et al.*'s studies, since, in addition to OLSR and Web-COBRA, we applied an Artificial Intelligence prediction method, namely Case-Based Reasoning (CBR) [1], and exploited a different validation method. In particular, we performed a *hold-out* cross by using 15 applications as training set and 10 further applications as test set. We applied this validation since it is considered theoretically the best option in specific cases, e.g., when using projects started after a certain date as hold-out, as in our case [16]. Moreover the split reflects the real outcoming of the software company development process since the observations included in the test set were developed after the ones included in the training set. Moreover, the 25 Web applications used in our study are more recent, thus exploiting newer technologies, development environments, etc.., and are much bigger than those used in [29].

The remainder of the paper is organized as follows. In Section 2 we report on the experimental method we exploited to establish whether Web Objects can be used to predict the development effort of Web applications. The results of the empirical analysis are reported and discussed in Section 3, while a discussion about the empirical study validity is presented in Section 4. Section 5 reports on the related work while Section 6 concludes the paper giving some final remarks.

2 Experimental Method

This section presents the design of the empirical study carried out to assess the effectiveness of Web Objects for sizing Web applications[2]. The research question we addressed is:

[RQ1] Can the Web Objects measure provide good estimations of Web applications development effort when used in combination with OLSR / CBR / Web-COBRA?

It is worth noting that our experimental settings allowed us to gain insight on two consequent research questions:

[RQ2] Are the estimates obtained using Web Objects statistically superior to the estimates obtained using Function Points?

---

[1] COBRA is a trademark of the Fraunhofer Institute - http://www.fraunhofer.de/

[2] Details on the design of the case study can be find in the technical report available at: http://www.dmi.unisa.it/people/gravino/www/work/Report_WO_Gravino2011-01-18/TechReport_WO_Gravino2011-01-18.pdf

[RQ3] Which estimation method, among OLSR, CBR, and Web-COBRA, provides the best predictions, when used in combination with Web Objects?

2.1 The Dataset

Data for our empirical study were provided by an Italian medium-sized software company, whose core business is the development of enterprise information systems, mainly for local and central government. Among its clients, there are health organizations, research centers, industries, and other public institutions. The company is specialized in the design, development, and management of solutions for Web portals, enterprise intranet/extranet applications (such as Content Management Systems, e-commerce, work-flow managers, etc.), and Geographical Information Systems. It has about fifty employees, it is certified ISO 9001:2000, and it is also a certified partner of Microsoft, Oracle, and ESRI.

The company first provided us data on 15 projects, and then data on further 10 applications. These two sets include e-government, e-banking, Web portals, and Intranet applications, developed between 2003 and 2008, and are quite homogeneous in terms of adopted technologies and development teams. In particular, all the projects were developed by exploiting SUN J2EE or Microsoft .NET technologies. Oracle has been the most commonly adopted DBMS, but also SQL Server, Access, and MySQL were employed in some of these projects.

Table 1 reports some summary statistics on these 25 projects, aggregated on the two datasets. The variables employed in our empirical analysis are *EFH*, i.e. the actual effort, expressed in terms of person/hours, *WO*, expressed in terms of number of Web Objects, and *FP*, expressed in terms of number of Function Points. Further details on how these data were collected are discussed in Section 4.

Table 1. Descriptive statistics of EFH, WO, and FP for the study

Dataset	Var	Min	Max	Mean	Median	Std. Dev.
I (15 observations)	EFH	1176	3712	2677.867	2792	827.115
	WO	465	2258	1464.867	1389	543.986
	FP	110	601	360.200	355	167.009
II (10 observations)	EFH	782	4537	2511.778	2498	1265.208
	WO	323	3078	1503.000	1271	960.927
	FP	175	973	459.600	327.5	273.612

2.2 The Web Objects Method

The Web Objects method was proposed by Reifer to measure the size of Web applications [26]. In particular, Reifer added four new Web-related components, namely Multimedia Files, Web Building Blocks, Scripts, and Links, to the five predictors of the FPA method. A detailed description of these components can be found in the Reifer "*white paper*" explaining the counting conventions of the Web Objects method [27].

To size a Web application accordingly to the method, a Measurer has to compute the Function Points in the traditional way. Then he/she has to identify the Web-related components that have not yet counted. Similarly to FPA, the further step is to

determine the complexity of the identified instances of the nine components. To support this task, Reifer provided a calculation worksheet in [65] that was subsequently modified by Ruhe [28]. We used this latter version, since we were interested in replicating Ruhe's studies. Thus, the application size in terms of Web Objects is obtained by adding the identified component instances taking into account the weights that are related to each component.

2.3 The Employed Effort Estimation Methods

Several techniques have been proposed in the literature to be employed for effort estimation [3]. In our empirical analysis, we applied OLSR [24][23] and Web-COBRA [29], since they have been applied in previous studies to assess the effectiveness of Web Objects in estimating Web application development effort [29][30]. Furthermore, we also employed CBR [1] since, together with OLSR, it is one of the most diffuse techniques in the industrial context and in several researches to estimate Web application development effort (see e.g., [10][11][19][21]).

OLSR. It is a statistical technique that explores the relationship between a dependent variable and one or more independent variables [24][23], providing a prediction model described by an equation

$$y = b_1x_1 + b_2x_2 + ... + b_nx_n + c \quad (1)$$

where y is the dependent variable, x_1, x_2, ..., x_n are the independent variables, b_i is the coefficient that represents the amount the variable y changes when the variables x_i changes 1 unit, and c is the intercept. In our empirical study we exploited OLSR to obtain a linear regression model that use the variable representing the effort as dependent (namely EFH) and the variable denoting the employed size measure (namely WO) as independent. Once the prediction model is constructed, the effort estimation for a new Web application is obtained by sizing the application in terms of Web Obejcts, and using this value in the obtained model.

Web-COBRA. It is an adaptation of COBRA, proposed to estimate the development effort of Web applications, taking into account "the needs of a typical Web application company" [29]. In the following we describe only the key aspects of this method; the interested reader can consult [28][29] for further details.

To apply Web-COBRA, two key aspects have to be setup for a specific environment:

1. The set of external factors that can lead to a rise of the cost for an application within the specific domain. These factors are modeled by introducing the concept of cost overhead, defined as "the additional percentage on top of the cost of an application running under optimal conditions" [4].
2. The relationship between cost overhead and effort.

The first aspect is captured by a *causal model*, i.e. a list of all the cost factors (and their relationships) that may affect a development cost within a specific domain. This conceptual, qualitative model is obtained through the acquisition of experts' knowledge. Then, the experts are asked to "quantify" the effect of each of these identified factors on the development effort, by specifying the percentage of overhead above an "optimal" application that each factor may induce. Since different experts may

provide different estimations of these percentages, basing on their previous experience, these factors are modeled as uncertain variables requiring a minimal, most likely, and maximal values. For example, experts may agree that the factor "safety of the Web application" may affect the development effort ranging from 10% (minimal), through 50% (most likely), to 80% (maximal). Then, a triangular distribution of these cost overheads is calculated. It is worth noting that the range of the distribution provides an indication on how uncertain the experts are about the overhead inducted by the specific cost factor [29].

As for the second step, the relationship between the cost overhead and the development effort is modeled by using the OLSR and employing past data of the company. The causal model and the determined relationship between effort and cost overhead are used to obtain the effort estimations for new applications. In this step a Monte Carlo simulation can be run to provide a distribution from where an estimate of the effort can be obtained by taking the mean of the distribution [29].

CBR. It is an Artificial Intelligence technique that allows us to predict the effort of a new Web application (target case) by considering some similar applications previously developed (case base) [1]. In particular once the applications are described in terms of some features (such as the size), the similarity between the target case and the others in the case base is measured, and the most similar ones are selected, possibly with adaptations, to obtain the estimation. To apply the method, a Measurer has to choose an appropriate similarity function, the number of analogies to select the projects to consider for the estimation, and the analogy adaptation strategy for generating the estimation. Some supporting tools can help doing these tasks.

2.4 Validation Method and Evaluation Criteria

In order to validate the obtained effort estimation models we performed a hold-out cross validation approach [16], employing datasets I and II of Table 1. Dataset I (training set) was used to train the effort estimation techniques while dataset II (test set) was used to validate the obtained models.

To assess the derived estimations, we used some summary measures, namely MMRE, MdMRE, and Pred(25) [8]. In the following, we briefly recall their main underlying concepts.

The *Magnitude of Relative Error* (MRE) [8] is defined as

$$MRE = |EFHreal - EFHpredl/EFHreal \quad (2)$$

where EFHreal and EFHpred are the actual and the predicted efforts, respectively. MRE has to be calculated for each observation in the test set. Then, the MRE values have to be aggregated across all the observations. We used the mean and the median, giving rise to the *Mean of MRE* (MMRE), and *Median of MRE* (MdMRE), where the latter is less sensitive to extreme values [20]. According to [8], a good effort prediction model should have a MMRE$\leq$0.25, to denote that the mean estimation error should be less than 25%.

The *Prediction at level l*%, also known as Pred(*l*), is another useful indicator that measures the percentage of estimates whose error is less than *l*%, where *l* is usually set at 25% [8]. It can be defined as

$$Pred(25) = k/N \quad (3)$$

where k is the number of observations whose MRE is less than or equal to 0.25, and N is the total number of observations. Again, according to [8], a good prediction approach should present a Pred(25) $\geq$ 0.75, meaning that at least 75% of the predicted values should fall within 25% of their actual values.

Moreover, we tested the statistical significance of the obtained results, by using absolute residuals, in order to establish if one of employed estimation measures provides significantly better results than the other [18][20]. In particular, we performed statistical tests (i.e., T-Test or Wilcoxon signed rank test when the distributions were not normally distributed) to verify the following null hypothesis: "the two considered populations have identical distributions". This kind of test is used to verify the hypothesis that the mean of the differences in the pairs is zero.

To have also an indication of the practical/managerial significance of the results we verified the effect size [15]. Effect size is a simple way of quantifying the difference between two groups. Employing the Wilcoxon test and the T-test, the effect sizes is determined by using the formula: r = Z-score/sqrt(N), where N is the number of observations. In particular, we first calculated the effect size and then compared it to the Cohen's benchmarks [6]: so r=0.20 indicates a small effect, r=0.50 indicates medium effect, and r=0.80 indicates a large effect.

Finally, as suggested in [21], we also analyzed the values of the summary statistics MMRE, MdMRE, and Pred(25) obtained by employing the mean effort (MeanEFH) and the median effort (MedianEFH) of the training set as estimated effort. Indeed, if the prediction accuracy obtained with complex measures/techniques is comparable with those got with the mean or median effort, then a software company could simply use the mean or the median effort of its past applications rather than dealing with complex computations of software sizes, such as Web Objects, to predict development effort.

3 Empirical Results

The following subsections present the results of the empirical analysis we carried out to establish whether the Web Objects measure is a good indicator of Web application development effort, when used in combination with OLSR, Web-COBRA, or CBR. As benchmark, we compared the predictions with those obtained with traditional Function Points.

3.1 Obtaining Estimates with OLSR

We performed the OLSR analysis to build the effort estimation model by using the training set of 15 Web applications (i.e., dataset I of Table 1). We applied OLSR two times: as independent variable we used in the first one WO, while in the second run FP. In both the cases, we preliminarily carried out an outlier's examination to remove potential extreme values which may influence the models, and then we verified the assumptions underlying the OLSR analysis. Table 2 shows the results of the OLSR applied with WO, in terms of R^2 (an indication of the goodness of the model), F-value and the corresponding p-value (denoted by Sign. F), whose high and low values,

respectively, denote a high degree of confidence for the estimation. Moreover, we performed a t statistic and determined the p-value and the t-value of the coefficient and the intercept for the obtained prediction model, to evaluate its statistical significance. A p-value lower than 0.05 indicates we can reject the null hypothesis that the variable is not significant in the considered model, with a confidence of 95%. As for the t-value, a variable is significant if the corresponding t-value is greater than 1.5. As we can see from Table 2, both the criteria are matched.

Table 2. The results of the OLSR analysis with WO

	Value	**Std. Err**	**t-value**	**p-value**
Coefficient	1.246	0.241	5.162	0.000
Intercept	851.912	375.814	2.267	0.041
R^2	**Adjusted R^2**	**Std. Err**	**F**	**Sign. F**
0.672	0.647	491.513	26.645	0.000

The results of the application of the OLSR with FP are reported in Table 3. Even if the coefficient and the intercept can be considered accurate and significant as from the t statistic, the R^2 and F values are lower than those obtained with WO, pointing out a weaker correlation between FP and EFH.

To understand the effectiveness of these models in predicting the development effort, their accuracy has been evaluated on a test set of 10 Web applications (i.e., dataset II of Table 1). The results are reported in Table 4.

Table 3. The results of the OLSR analysis with FP

	Value	**Std. Err**	**t-value**	**p-value**
Coefficient	3.853	0.863	4.464	0.001
Intercept	1290.121	340.651	3.787	0.002
R^2	**Adjusted R^2**	**Std. Err**	**F**	**Sign. F**
0.605	0.575	539.3	19.93	0.001

Based on the commonly accepted thresholds provided in [8], even if the value of Pred(25) is slightly less than 0.75, we can conclude that WO is a good indicator of Web application development effort, when used in combination with OLSR. Furthermore, we can note that the estimates obtained using WO are much better than those obtained with FP, with about half the mean and median error. Also the T-test confirmed the superiority of WO, highlighting that their estimations are significantly better than those obtained with FP (p-value=0.008). Finally we computed the effect size, whose analysis revealed a medium effect size (r=0.54), according to the widely used Cohen's benchmarks [6].

Table 4. The results of OLSR

	MMRE	**MdMRE**	**Pred(25)**
OLSR with WO	0.21	0.15	0.70
OLSR with FP	0.46	0.28	0.40

3.2 Obtaining Estimates with Web-COBRA

To apply the Web-COBRA method, the following steps were conducted:

1) Identification and quantification of cost factors.
2) Data collection for the Web applications involved in the case study.

As for 1) it is worth noting that a large number of cost drivers may affect the development cost of software applications. However, for each domain, only a subset of these factors turns out to be relevant [4][29]. We drafted an initial list including the cost factors identified in [28][29] that was submitted to five experts of the software company involved in our empirical study. Then a Delphi method [18] was adopted until they agreed on the final set of cost drivers. They were asked to comment, basing on their experience, on the clarity of the factors (to avoid that different project managers could interpret them in different ways), on their completeness (to avoid that some key factors might not be considered), and on relevance for the Web application development domain, working also to reduce as much as possible redundancies and overlaps. A final list of 10 cost drivers was devised. They are reported in Table 5. It is worth noting that this list includes four cost factors employed by Ruhe *et al.* in [28][29]: Novelty of Requirements, Importance of Software Reliability, Novelty of Technology, and Developer's Technical Capabilities[3]. Then, the experts were asked to quantify the cost factors, specifying their minimal, most likely, and maximal inducted overhead (see Table 5). Again, a Delphi method was used to obtain a single representative triple for each cost factor. Subsequently, for each Web application p, the corresponding project manager specified the influence of the cost factors on p by a value in the range 0..3, where 0 means that no influence was due to that factor, and 3 represents the highest impact. Thus, the information on the cost overhead for each project p was obtained by the sum of all the triangular distributions of cost factors specified for p, taking into account their minimal, most likely, and maximal values of Table 6.

Table 5. Identified cost factors and their influence

Cost Factor	Minimal	Most Likely	Maximal
Novelty of Requirements (CF1)	10%	35%	70%
Importance of Software Portability (CF2)	7%	25%	60%
Importance of Software Reliability (CF3)	5%	20%	60%
Importance of Software Usability (CF4)	7%	30%	65%
Importance of Software Efficiency and Performance (CF5)	7%	20%	50%
Novelty of Technologies (CF6)	5%	25%	65%
Integration/Interaction with legacy systems (CF7)	20%	35%	70%
Temporal Overlap with other projects (CF8)	10%	35%	60%
Productivity of the adopted technological platform (CF9)	15%	45%	65%
Developer's Technical Capabilities (CF10)	10%	35%	65%

[3] Importance of Software Reliability was not included in the final list selected by the project managers in the experiment presented in [29].

The information on Effort (namely EFH), Size (expressed in terms of WO or FP), and *co_overhead* obtained from cost factors was exploited to build a model and validate it. Observe that Web-COBRA assumes that the relationship between effort and size is linear [29]. We have performed the required statistical tests to verify this linearity in our dataset. The obtained equation is:

$$Effort = 0.477 \cdot WO^{*}co\_overhead + 1095.89 \quad (4)$$

Moreover, the size of a Web application is modeled as an uncertain variable, which underlies a triangular distribution and an uncertainty of 5% was considered in [29]. Then, we applied a hold-out cross validation, by employing the training and the test sets in Table I. Moreover, we run a Monte Carlo simulation (considering 1000 iterations) that allowed us to use the relationship between cost overhead and effort together with the causal model to obtain a probability distribution of the effort for the new project [29]. Then, the mean value of the distribution was used as the estimated effort value. Table 6 shows the results of the validation we obtained in terms of MMRE, MdMRE, and Pred(25), by applying Web-COBRA in combination with WO and FP (this latter analysis was not performed by Ruhe *et al.* in [29]).

Again we got a superiority of WO, whose predictions fit the acceptable threshold defined in [8]. This does not hold for FP. Also statistical tests highlight that the estimates obtained with WO are significantly better than those obtained with FP (p-value=0.003) with a medium effect size (r=0.71).

Table 6. The results of Web-COBRA

	MMRE	**MdMRE**	**Pred(25)**
Web-COBRA with WO	0.18	0.12	0.80
Web-COBRA with FP	0.29	0.25	0.50

3.3 Obtaining Estimates with CBR

To apply CBR, in our empirical study we exploited the tool *ANGEL* [31] [30]. It implements the Euclidean distance as similarity function, using variables normalized between 0 and 1, and allows users to choose the relevant features, the number of analogies, and the analogy adaptation technique for generating the estimations. Since we dealt with a not so large dataset, we used 1, 2, and 3 analogies, as suggested in many similar works [20]. To obtain the estimation, once the most similar cases were determined (exploiting information on the size, i.e., Web Objects), we employed three widely adopted adaptation strategies: the mean of k analogies (simple average), the inverse distance weighted mean [20], and the inverse rank weighted mean [31]. So, we obtained 10 estimations and the corresponding residuals, for each selection of the number of analogies and of the analogy adaptation techniques. Since we carried out a hold-out cross validation, each estimation was obtained by selecting a target observation from the test dataset, and by considering as case base the observations in the training dataset. Table 7 shows the best results in terms of MMRE, MdMRE, and Pred(25), for both WO and FP. These results are the best we got, being obtained by employing 2 analogies and the mean of k analogies as adaptation strategy.

Table 7. The results of CBR using ANGEL

	MMRE	MdMRE	Pred(25)
CBR with WO	0.22	0.12	0.70
CBR with FP	0.49	0.17	0.60

As for OLSR and Web-COBRA, WO outperformed FP also with CBR. In particular, the MMRE and MdMRE values satisfy the usual acceptance thresholds of [8], while Pred(25) value is slightly less than 0.75. In contrast with the results achieved with OLSR and Web-COBRA, the statistical tests revealed that the estimates with WO are not significantly superior to those obtained with FP (p-value = 0.072), with a small effect size (r=0.42).

4 Discussion and Comparison

In this section we discuss the results we have gathered and compare them with those achieved by Ruhe *et al.* in [29][30].

The MMRE, MdMRE, and Pred(25) values reported in Table 4, Table 6, and Table 7 suggest that the Web Objects measure is a good indicator of Web application size, when used in combination with the prediction techniques we considered. These results also highlight that Web Objects outperforms Function Points in terms of prediction accuracy. This is a confirmation to an expected result, since the Web Objects method was conceived to overcome the limitations of FPA when dealing with Web applications. Moreover, we can observe that Web-COBRA provided slightly better results than OLSR and CBR, in terms of MMRE, MdMRE, and Pred(25).

The above results corroborate what suggested by the common sense: Web-COBRA, taking into account also many non-functional aspects of the software process and product, provides improved estimations than the two other techniques relying only on the Web Objects size measure. On the other hand, it is very interesting to point out that Web-COBRA applied with FP provided worse results than OLSR with WO. This means that the four new components sized by the Web Objects method are much more correlated to the effort than the non-functional factors handled by Web-COBRA. This is also confirmed by the fact that there is no statistically significant difference between the three techniques.

As designed, we compared the predictions with those obtained by the simple mean or median of the effort of the whole training set. These predictions are very poor, as reported in Table 8, since they do not satisfy the typical acceptance thresholds [8]. Moreover, predictions obtained with WO and FP based models are significantly better than those obtained using MeanEFH and MedianEFH.

Table 8. The results of MeanEFH and MedianEFH

	MMRE	MdMRE	Pred(25)
MeanEFH	0.63	0.37	0.40
MedianEFH	0.68	0.34	0.40

Summarizing, regarding the research questions RQ1, RQ2, and RQ3, the results of the performed empirical analysis suggest that:

[RQ1] The Web Objects measure resulted to be a good indicator of Web application development effort, when used in combination with OLSR, CBR, and Web-COBRA, since the values of summary measures are very close or match the thresholds usually adopted in this domain [8].

[RQ2] The estimates obtained with Web Objects turned out to be statistically superior to the ones achieved with Function Points in combination with OLSR and Web-COBRA.

[RQ3] Even if Web-COBRA provided slightly better results than OLSR and CBR in terms of summary measures there is no statistically significant difference in the estimations obtained by applying the three methods in combination with Web Objects.

It is worth mentioning that the present study confirmed and extended two our previous studies employing a different validation method and using a larger dataset. In particular, in [13] we assessed the effectiveness of Web Objects as indicators of development effort, when used in combination with OLSR, by employing dataset I of Table 1 (of 15 Web applications) as training set and further 4 Web applications as test set. The results revealed that the Web Objects measure is a good indicator of the development effort since we obtained MMRE=0.14, MdMRE=0.06, and Pred(25)=0.75. Moreover, in [12] we assessed the use of Web Objects in combination with Web-COBRA, using only dataset I of Table 1, with a leave-1-out cross validation, obtaining MMRE=0.11, MdMRE=0.10, and Pred(25)=0.93.

4.1 Comparison with Ruhe *et. al.* Analyses

Ruhe *et al.* [29][30] carried out empirical analyses based on a dataset of 12 Web applications developed between 1998 and 2002 by an Australian software company, with about twenty employees. The most of these projects were new developments, even if there were also enhancements, and re-development projects. The Web Objects measure was used as size metrics in combination with OLSR and Web-COBRA and a leave-1-out cross validation was exploited to validate the obtained estimation techniques. Ruhe *et al.* also employed summary measures MMRE, MdMRE, and Pred(25) and statistical test (T-test) to evaluate the accuracy of the obtained estimates.

Table 9 reports on the values of the summary statistics on the estimation accuracy obtained in [29][30]. We can observe that the summary values we obtained in our empirical analyses are slightly better than those obtained by Ruhe *et al.* Thus, the study reported in the present paper is confirming the results of the previous researches showing the effectiveness of Web Objects. Moreover, in all the three studies, the performed statistical tests (i.e., T-test) revealed that the estimates achieved with Web Objects significantly outperformed the estimates obtained with Function Points. As for comparison of the employed estimation techniques, the statistical analysis also suggested that the estimates obtained with OLSR and Web-COBRA are comparable, i.e., there is no significant difference between them.

Table 9. Ruhe *et al.'s* results reported in [29][30]

	MMRE	MdMRE	Pred(25)
OLSR with FP	0.33	0.33	0.42
OLSR with WO	0.24	0.23	0.67
Web-COBRA with WO	0.17	0.15	0.75

The results we obtained extend the ones of Ruhe *et al.* in several aspects. Indeed, besides the techniques employed in their case study, we also exploited CBR, still obtaining good results, thus showing a sort of robustness of Web Objects with respect to the employed techniques. As for the performed empirical analysis, we exploited further benchmarks (i.e., MeanEFH and MedianEFH) and more tests (i.e., effect size).

From a managerial point of view, our results extend the ones provided by Ruhe *et al.*, showing the scalability of the Web Objects measure, in terms of technologies and size of the considered projects. Indeed, the 25 Web applications used in our empirical study are more recent, being developed between 2003 and 2008, thus exploiting newer technologies, development environments, etc.. Moreover, they are much bigger than those used in [29]. Table 10 provides some descriptive statistics about the set of Web applications we employed in our case study and the dataset considered by Ruhe *et al.* in their study [29]. In particular, we reported on the size, the actual effort (in terms of person/hours), and the peak staff. We can observe that the mean effort of our dataset is about three times the one of the dataset used in [29] and applications are characterized also by a bigger size in terms of Web Objects (about five times bigger than those in [29]). It is interesting to note that the number of Function Points is not so different among the two datasets, since in our case the applications are about 1.5 times bigger than those of Ruhe *et al*., in terms of this size measure. A possible interpretation we gave to this phenomenon is that our applications highly exploit Web Building Blocks and Multimedia elements, which are considered by the Web Objects method but not by the FPA method.

Table 10. Descriptive statistics of EFH, WO, and Peak Staff

Our study					
	Min	**Max**	**Median**	**Mean**	**Std. Dev.**
WO	323	3,078	1366	1480	720.602
EFH (person/hours)	782	4537	2686	2577	988.136
Peak Staff	6	7	6	6.2	0.4
Ruhe *et al.*'s study					
	Min	**Max**	**Median**	**Mean**	**Std. Dev.**
WO	67	792	Unknown	284	227
EFH (person/hours)	267	2,504	Unknown	883	710
Peak Staff	2	6	Unknown	3	1.5

5 The Empirical Study Validity

It is widely recognized that several factors can bias the construct, internal, external, and conclusion validity of empirical studies [17] [20]. As for the construct validity, the choice of the size measure and how to collect information to determine size

measure and actual effort represent crucial aspects. Regarding the selection of the size measure, we employed a solution specifically proposed for Web applications by Reifer [26] and used in the previous case studies we have replicated. The software company uses timesheets to keep track of effort information, where each team member annotates his/her development effort and weekly each project manager stores the sum of the efforts for the team. To calculate the size measure, the authors defined a form to be filled in by the project managers. To apply the Web Objects method they employed the counting conventions of the FPA method [14] and followed the suggestions provided by Reifer in his "Web Objects White Paper" [27]. One of the authors analyzed the filled forms in order to cross-check the provided information. The same author calculated the values of the size measure. As for the collection of the information on the cost factors, we defined a questionnaire together with some company experts. Then, this questionnaire was submitted to the project managers of the considered Web applications. Thus, the data collection task was carried in a controlled and uniform fashion, making us confident on the accuracy of the results.

With regards to internal validity no initial selection of the subjects was carried out, so no bias has been apparently introduced. Moreover, the Web applications were developed with technologies and methods that subjects had experienced. Consequently, confounding effects from the employed methods and tools can be excluded. Moreover, the questionnaires used were the same for all the Web applications and the project managers were instructed on how to use them. Instrumentation effects in general did not occur in this kind of studies. As for the conclusion validity we carefully applied the statistical tests, verifying all the required assumptions. Biases about external validity were mitigated by considering as dataset a representative sample of modern Web applications. However, it is recognized that the results obtained in an industrial context might not hold in other contexts. Indeed, each context might be characterized by some specific project and human factors, such as development process, developer experience, application domain, tools, technologies used, time, and budget constraints [5].

6 Related Work

Besides the Web-COBRA method and the Web Objects measure, other estimation techniques and size measures have been proposed in the literature to be employed for estimating Web applications development effort.

The COSMIC [9] method has been applied to Web applications by some researchers in the last years [10][19]. In particular, Mendes *et al*. applied it to 37 Web systems developed by academic students, by constructing an effort estimation model with OLSR [19]. Unfortunately, this model did not provide good estimations and replications of the empirical study were highly recommended. Subsequently, an empirical study based on the use of 44 Web applications developed by academic students, was performed to assess the COSMIC approach [10]. The effort estimation model obtained by employing the OLSR provided encouraging results.

Some authors investigated the usefulness of size measures specific for Web applications such as number of Web pages, media elements, internal links, etc., [11], [20]. Among them, Mendes *et al*. also built the Tukutuku database [21], which aims to collect data from completed Web sites and applications to develop Web cost estimation models and to benchmark productivity across and within Web Companies. Several studies were conducted to investigate and compare the effectiveness of these measures in combination with estimation techniques like OLSR, CBR, Regression Tree (RT), and Bayesian Networks (BN). In particular, in [20] a dataset of 37 Web systems developed by academic students was exploited and the empirical results suggested that Stepwise Regression (SWR) provided statistically significant superior predictions than the other techniques when using length size measures, such as number of Web pages, number of new media. On the contrary, a study exploiting a dataset containing data on 15 Web software applications developed by a single Web company (the ones also employed in the empirical study presented in this paper) revealed that none of the employed techniques (i.e., SWR, RT, and CBR was statistically significantly superior than others [11]. Recently, Mendes and Mosley investigated the use of Bayesian Networks for Web effort estimation using the Web applications of the Tukutuku database [22]. In particular, they employed two training sets, each with 130 Web applications, to construct the models while their accuracy was measured using two test sets, each containing data on 65 Web applications. The analysis revealed that Manual SWR provided significantly better estimations than any of the models obtained by using Bayesian Networks and is the only approach that provided significantly better results than the median effort based model.

7 Conclusions

In this paper, we investigated the effectiveness of the Web Objects measure as indicator of Web application development effort. In particular, we replicated the two studies carried out by Ruhe *et al.* [29]. The contribution of our work to the body of knowledge can be summarized as in the following:

- we confirmed the effectiveness of the Web Objects measure as indicator of Web application development effort, when used in combination with OLSR and Web-COBRA, and verified that this holds also using CBR;
- we confirmed that the Web Objects method provides statistically superior results than the FPA method when used in combination with OLSR and Web-COBRA;
- we showed that there are no statistically significant differences in the results obtained with OLSR, CBR, and Web-COBRA, i.e., the approaches are comparable when using the Web Objects measure.

Of course, the experimental results here presented hold only with respect to the dataset took into account and they should be assessed on further data as soon as they are available. However, they are surely interesting enough to suggest the use of the Web Objects measure as indicator of Web application development effort, also because confirm the results of Ruhe *et al*.. In the future, we intend to further assess Web Objects by considering a different context.

References

[1] Aamodt, A., Plaza, E.: Case-based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communication 7(1), 39–59 (1994)

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

[3] Briand, L., Wieczorek, I.: Software resource estimation. In: Encyclopedia of Software Engineering, pp. 1160–1196 (2002)

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

[5] Briand, L.C., Wust, J.: Modeling Development Effort in Object-Oriented Systems Using Design Properties. IEEE Transactions on Software Engineering 27(11), 963–986 (2001)

[6] Cohen, J.: Statistical power analysis for the behavioral science. Lawrence Erlbaum, Hillsdale (1998)

[7] Conallen, J.: Building Web Applications with UML. Addison-Wesley, Reading (1999)

[8] Conte, D., Dunsmore, H., Shen, V.: Software Engineering Metrics and Models. The Benjamin/Cummings Publishing Company, Inc. (1986)

[9] COSMIC (2007), `http://www.cosmicon.com`

[10] Costagliola, G., Di Martino, S., Ferrucci, F., Gravino, C., Tortora, G., Vitiello, G.: A COSMIC-FFP approach to Predict Web Application Development Effort. Journal of Web Engineering 5(2) (2006)

[11] Costagliola, G., Di Martino, S., Ferrucci, F., Gravino, C., Tortora, G., Vitiello, G.: Effort Estimation Modeling Techniques: A Case Study for Web Applications. In: Proceedings of the International Conference on Web Engineering, pp. 161–165. ACM Press, New York (2006)

[12] Di Martino, S., Ferrucci, F., Gravino, C.: An Empirical Study on the Use of Web-COBRA and Web Objects to Estimate Web Application Development Effort. In: Gaedke, M., Grossniklaus, M., Díaz, O. (eds.) ICWE 2009. LNCS, vol. 5648, pp. 213–220. Springer, Heidelberg (2009)

[13] Ferrucci, F., Gravino, C., Di Martino, S.: A Case Study Using Web Objects and COSMIC for Effort Estimation of Web Applications. In: Proceedings of Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2008), pp. 441–448 (2008)

[14] I. F. P. U. G., Function point counting practices manual, release 4.2.1

[15] Kampenes, V., Dyba, T., Hannay, J., Sjoberg, D.: A systematic review of effect size in software engineering experiments. Information & Software Technology 49(11-12), 1073–1086 (2007)

[16] Kitchenham, B., Mendes, E., Travassos: Cross versus Within-Company Cost Estimation Studies: A systematic Review. IEEE Transactions on Software Engineering 33(5), 316–329 (2007)

[17] Kitchenham, B., Pickard, L., Pfleeger, S.L.: Case Studies for Method and Tool Evaluation. IEEE Software 12(4), 52–62 (1995)

[18] Kitchenham, B., Pickard, L.M., MacDonell, S.G., Shepperd, M.J.: What accuracy statistics really measure. IEE Proceedings Software 148(3), 81–85 (2001)

[19] Mendes, E., Counsell, S., Mosley, N.: Comparison of Web Size Measures for Predicting Web Design and Authoring Effort. IEE Proceedings-Software 149(3), 86–92 (2002)

[20] Mendes, E., Counsell, S., Mosley, N., Triggs, C., Watson, I.: A Comparative Study of Cost Estimation Models for Web Hypermedia Applications. Empirical Software Engineering 8(23) (2003)

[21] Mendes, E., Kitchenham, B.: Further Comparison of Cross-company and Within-company Effort Estimation Models for Web Applications. In: Proceedings of International Software Metrics Symposium, pp. 348–357. IEEE press, Los Alamitos (2004)
[22] Mendes, E., Mosley, N.: Bayesian Network Models for Web Effort Prediction: A Comparative Study. IEEE Transactions on Software Engineering 34(6), 723–737 (2008)
[23] Mendes, E., Mosley, N., Counsell, S.: Investigating Web Size Metrics for Early Web Cost Estimation. Journal of Systems and Software 77(2), 157–172 (2005)
[24] Montgomery, D., Peck, E., Vining, G.: *Introduction to Linear Regression Analysis*. John Wiley and Sons, Inc., Chichester (1986)
[25] Morisio, M., Stamelos, I., Spahos, V., Romano, D.: Measuring Functionality and Productivity in Web-based applications: a Case Study. In: Proceedings of the International Software Metrics Symposium, pp. 111–118. IEEE press, Los Alamitos (1999)
[26] Reifer, D.: Web-Development: Estimating Quick-Time-to-Market Software. IEEE Software 17(8), 57–64 (2000)
[27] Reifer, D.: Web Objects Counting Conventions, Reifer Consultants (March 2001), `http://www.reifer.com/download.html`
[28] Ruhe, M.: The Accurate and Early Effort Estimation of Web Applications, PhD Thesis, Fraunhofer IESE 2002 (2002)
[29] Ruhe, M., Jeffery, R., Wieczorek, I.: Cost estimation for Web applications. In: Proceedings of International Conference on Software Engineering, pp. 285–294. IEEE press, Los Alamitos (2003)
[30] Ruhe, M., Jeffery, R., Wieczorek, I.: Using Web Objects for Estimating Software Development Effort for Web Applications. In: Proceedings of the International Software Metrics Symposium (2003)
[31] Shepperd, M., Schofield, C.: Estimating software Project Effort using Analogies. IEEE Transactions on Software Engineering 23(11), 736–743 (2000)