# An Empirical Study on the Use of Web-COBRA and Web Objects to Estimate Web Application Development Effort

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

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

**Abstract.** We have performed a replication of a previous study in order to further assess the effectiveness of Web-COBRA method, with the Web Objects measure, in predicting Web application development effort. The results of the empirical analysis confirm the interesting results of the previous study.

**Keywords:** Effort estimation method, Web-COBRA, Web applications.

## 1 Introduction

In the field of Web Engineering many techniques have been proposed to predict the effort required for the development of an application, and among them Web-COBRA is of particular interest, thanks to its ability to combine experts knowledge with a formal estimation model [21]. Web-COBRA is an adaptation of COBRA [4] and the key issue of these two methods is to exploit experts' knowledge, gathered in a controlled way, for identifying the main factors that can influence the development cost for a specific software company/domain, and this information is used to "adjust" the estimations coming from a model that exploits a size measure as cost driver.

To date, the effectiveness of Web-COBRA has been assessed only in one previous study, which exploited a dataset of 12 Web applications, sized using Web Objects, a measure proposed by Reifer specifically for Web applications [20]. Web Objects add four new Web-related components to the five function types of the Function Point Analysis method [10], namely Multimedia Files, Web Building Blocks, Scripts, and Links. The analysis performed in [21] provided interesting results and encouraged us to further analyze the effectiveness of Web-COBRA in combination with Web Objects. In particular, we have replicated that study in a different context, by considering a dataset of 15 bigger Web applications developed by an Italian software company.

The remainder of the paper is organized as follows. In Section 2 we report on the empirical analysis we performed together with a discussion of the gathered results. Section 3 contains a description of related work, while Section 4 concludes the paper giving final remarks and suggestions on future work.

M. Gaedke, M. Grossniklaus, and O. Díaz (Eds.): ICWE 2009, LNCS 5648, pp. 213 – 220, 2009.

## 2 Empirical Study

We start by describing the dataset employed in the empirical study and the differences with respect the dataset used in the previous analysis. Then, we report on the empirical analysis and discuss the results.

### 2.1 Dataset Description

The empirical study we present in this paper is based on a dataset coming from a medium-sized software company operating in Italy, whose core business is the development of enterprise information systems, mainly for local and central government. The company has about fifty employees and a turnover of about 5M €. It is certified ISO 9001:2000, and it is also a certified partner of Microsoft and Oracle.

Data used in the study are related to a set of 15 Web applications, including e-government, e-banking, Web portals, and Intranet applications. They have been developed by exploiting different Web-oriented technologies, such as J2EE, ASP.NET, etc... Oracle has been the most commonly adopted DBMS, but also SQL Server, Access and MySQL have been employed in some applications.

Table 1 provides some descriptive statistics about the considered dataset, taking into account the size (in terms of Web Objects), the actual effort (in terms of person-hours), and the peak staff (in terms of people involved in the project). Moreover, it also reports the statistics of the dataset considered by Ruhe *et al.* for the first assessment of Web-COBRA [21]. They used 12 Web applications, such as B2B, intranet or financial, developed between 1998 and 2002 by a small Australian software development company, with about twenty employees. The most of these projects were new developments, even if there were also enhancements, and re-development projects.

**Table 1.** Descriptive statistics of effort (in terms of person/hours), and size (in terms of Web Objects), and peak staff

| | **Min** | **Max** | **Median** | **Mean** | **Std. Dev.** |
|---|---|---|---|---|---|
| **Current study** | | | | | |
| *Web Objects* | 465 | 2,258 | 611 | 1,474.87 | 543.42 |
| *Effort* | 1176 | 3,712 | 1389 | 2,677.87 | 827.12 |
| *Peak Staff* | 6 | 7 | 6 | 6.2 | 0.4 |
| **Ruhe *et al.*'s study** | | | | | |
| *Web Objects* | 67 | 792 | Un-Know | 284 | 227 |
| *Effort* | 267 | 2,504 | Un-Know | 883 | 710 |
| *Peak Staff* | 2 | 6 | Un-Know | 3 | 1.5 |

The 15 Web applications considered in our replication are of the same kind, but they are more recent, being developed between 2003 and 2006. Other differences are related to the size of the applications and the size of the team. Indeed, in mean, the applications considered in this replication are 5.2 times bigger than the ones of Ruhe *et al.*, in terms of Web Objects. Moreover, the development teams considered in this replication are in mean 2 times bigger than the ones of Ruhe *et al.*

## 2.2 The Empirical Analysis

In order to apply the Web-COBRA method, we first identified and quantified the cost factors and then we collected data from Web application involved in our case study.

Regarding the identification and quantification of cost factors, a large number of cost drivers may affect the development cost of software applications. To select the set of the cost drivers that result significant for the considered domain [4], [21], we drafted an initial list, basing on a review of the literature on Web effort/cost estimation, including the cost factors identified in [21]. This preliminary list was submitted to five experts of the software company involved in our empirical study. Then a Delphi method [14] 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 2. It is worth noting that this list includes four cost factors employed by Ruhe *et al.* in [21]: Novelty of Requirements, Importance of Software Reliability, Novelty of Technology, and Developer's Technical Capabilities. Then, the experts were asked to quantify the cost factors, specifying their minimal, most likely, and maximal inducted overhead (see Table 2). Again, a Delphi method was used to obtain a single representative triple for each cost factor.

**Table 2.** 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% |

In the second step, the relationship between the cost overhead and the development cost was modeled by using the Ordinary Least Squares Regression (OLSR) and employing past data of the company. The information on the cost overhead for each project $p$ was obtained by the sum of all the triangular distributions of cost factors of $p$, given their minimal, most likely, and maximal values indicated by project managers. In particular, for each cost factor, the project manager specified its influence on the Web application by means of a value in the range 0..3, where 0 means that no influence was due to that factor, and 3 represents the highest impact. The descriptive statistics of the cost factors are shown in Table 3.

**Table 3.** Descriptive Statistics of cost factors

| | Min. | Max. | Mean | Median | Std. Dev. |
|---|---|---|---|---|---|
| *CF1* | 0 | 2 | 0.933 | 1 | 0.700 |
| *CF2* | 0 | 1 | 0.333 | 0 | 0.488 |
| *CF3* | 0 | 3 | 0.533 | 0 | 0.915 |
| *CF4* | 0 | 3 | 1.067 | 1 | 1.033 |
| *CF5* | 0 | 1 | 0.133 | 0 | 0.352 |
| *CF6* | 0 | 3 | 0.867 | 0 | 1.125 |
| *CF7* | 0 | 3 | 1.400 | 1 | 0.828 |
| *CF8* | 1 | 3 | 1.800 | 2 | 0.676 |
| *CF9* | 1 | 3 | 1.800 | 2 | 0.775 |
| *CF10* | 1 | 1 | 1.000 | 1 | 0 |

The information on Effort, Size (expressed in terms of Web Objects), and co_overhead was then exploited to build an estimation model and validate it. Observe that Web-COBRA assumes that the relationship between effort and size is linear [21]. We performed the required statistical tests to verify this linearity in our dataset. Moreover, the size of a Web application was modeled as an uncertain variable, which underlies a triangular distribution, and an uncertainty of 5% was considered in [21]. Then, we applied a leave-1-out cross validation, by performing 15 steps, where at each step the original dataset was partitioned into a training set of 14 Web applications and a validation set consisting of the remaining application. The training set was used to determine the estimation model and the validation set to assess the obtained estimates. Moreover, we ran a Monte Carlo simulation (considering 1000 iterations) that allowed us to use the relationship between cost overhead and effort together with the causal model in order to obtain a probability distribution of the effort for the new project [21]. Then, the mean value of the distribution was used as the estimated effort value.

To evaluate the accuracy of the obtained estimations, we employed some widely used summary measures, namely MMRE, MdMRE, and Pred(0.25) [6], together with boxplots of absolute residuals [13]. Table 4 contains the results we obtained, which highly fit the acceptable thresholds defined in [6] since MMRE (and MdMRE) values are less than 0.25 and Pred(0.25) value is greater than 0.75. This means that the mean error of the estimates we obtained is less than 25%, and that at least 75% of the estimated values fall within 25% of their actual values.

**Table 4.** Descriptive accuracy evaluation

| | MMRE | MdMRE | Pred(0.25) |
|---|---|---|---|
| *Web-COBRA with Web Objects* | 0.14 | 0.11 | 0.87 |
| *MeanEFH* | 0.34 | 0.27 | 0.47 |
| *MedianEFH* | 0.33 | 0.24 | 0.60 |

As suggested by Mendes and Kitchenham [18], we also compared MMRE, MdMRE, and Pred(0.25) with the mean of effort (i.e., MeanEFH) and the median of effort (i.e., MedianEFH) across all the dataset of past projects, as estimated value. The aim is to have a benchmark to assess whether the estimates obtained with Web-COBRA are significantly better than the trivial estimates based on the mean or median effort of all the previous projects. Indeed, if the estimates obtained with mean or median effort are similar to those obtained with the employed method then for the software company it could be more useful to simply use the mean or the median effort of the past developed applications rather than dealing with complex estimation techniques [18]. The results in Table 4 reveal that the values of MMRE and Pred(0.25) for MeanEFH and MedianEFH do not match the acceptance thresholds suggested in [6]. The boxplots of absolute residuals depicted in Figure 1 confirm that the estimations achieved by using Web-COBRA with Web Objects are better than those obtained using MeanEFH and MedianEFH. Indeed, the box length and the tails of the boxplot for Web-COBRA are smaller than those of boxplots for MeanEFH and MedianEFH. Furthermore, the median of boxplot for Web-COBRA is more close to zero than the other two boxplots.

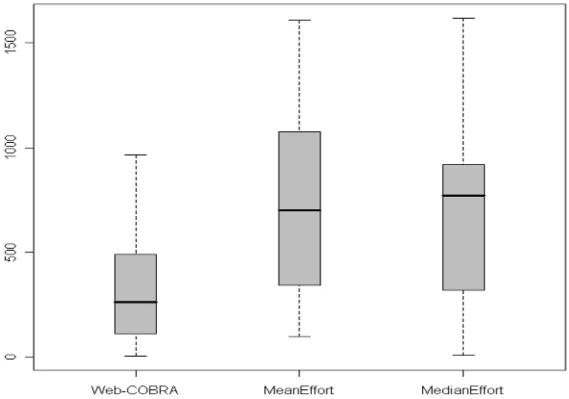

**Fig. 1.** The boxplots of absolute residuals

As suggested in [13], we also tested the statistical significance of the results obtained from the proposed models by using paired absolute residuals. To this end, we performed both the T-Test and the Wilcoxon test to verify the following null hypothesis: “the two considered population have identical distributions”. The analysis revealed that the absolute residuals obtained using Web-COBRA and Web Objects were significantly better than those obtained using MeanEFH and MedianEFH.

To have an indication of the practical/managerial significance of these results we analyzed the effect size, which has many advantages over the use of tests of statistical significance alone since “whereas p-values reveal whether a finding is statistically significant, effect size indicates practical significance” [11]. Employing the Wilcoxon test and the T-test, the effect sizes was determined by using the formula: r = Z-score / sqrt(N), where N is the number of observations. The statistics on effect size revealed that all results statistically significant were also practical significant according to the widely used Cohen's benchmarks [5]. Indeed, medium effect sizes were highlighted (i.e., 50 < r <0.80).

### 2.3 Comparison with the Previous Case Study

As for comparison with the previous empirical study, in Table 5 we have reported the values of the summary statistics on the prediction accuracy obtained in the previous research on Web-COBRA [21]. We can observe that the MMRE and Pred(0.25) values we obtained are slightly better than those of [21]. Thus, the analysis reported in the present paper has confirmed the results of [21] showing the effectiveness of Web-COBRA in combination with Web Objects to estimate Web application development effort. Moreover, our results extend the ones in [21], showing the scalability of the method. Indeed, as discussed in Section 2.1 the Web applications used in our empirical study have larger sizes than the Web applications used in [21]. Thus, our results suggest that Web-COBRA and Web Objects turn out to be good for estimating larger Web applications, too.

**Table 5.** Comparison with the results of [21]

| | **MMRE** | **MdMRE** | **Pred(0.25)** |
|---|---|---|---|
| *Our result* | 0.14 | 0.11 | 0.87 |
| *Ruhe et al. result* | 0.17 | 0.15 | 0.75 |

## 3 Related Work

Besides Web-COBRA method and Web Objects measure, other approaches were proposed in the literature for estimating Web applications development effort.

The COSMIC method [7] has been applied to Web applications by some researchers in the last years [9], [15], who exploited as estimation technique the OLSR. In particular, Mendes *et al.* applied the COSMIC approach to Web sites, i.e., without server-side elaborations [15]. Using data about 37 Web systems developed by academic students, they constructed an effort estimation model by applying OLSR. However, this model did not provide good estimations and replications of the study were highly recommended by Mendes *et al.* to find possible biases in the application of the method. Subsequently, the observation that dynamic Web applications are characterized by data movements (from a Web server to the client browser) suggested to apply the principles of the COSMIC method to size this type of Web applications [9]. An empirical study based on the use of 44 Web applications developed by academic students, was performed to assess the COSMIC approach [9]. 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. [1], [3], [8], [15], [16], [17]. Several studies were conducted to investigate and compare the effectiveness of these measures in combination with estimation techniques like Linear and Stepwise Regression, Case-Based Reasoning, Regression Tree, and Bayesian Networks [2], [3], [8], [16], [17], [19]. In particular, in [16] a dataset of 37

Web systems developed by academic students was exploited and the empirical results suggested that Stepwise Regression provided statistically significant superior predictions than the other techniques when using length size measures, such as number of Web pages, number of new media. By employing the same dataset, in [17] the Regression Tree gave worse results than Stepwise Regression and Case-Based Reasoning, and the authors highlighted that the models obtained with Linear and Stepwise Regression generally gave statistically significant better results than Case-Based Reasoning and Regression Tree. 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., Stepwise Regression, Regression Tree, and Case-Based Reasoning) was statistically significantly superior than others [8]. Recently, Mendes and Mosley investigated the use of Bayesian Networks for Web effort estimation using the Web applications of the Tukutuku database [19]. 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 Stepwise Regression 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.

Baresi and Morasca proposed a different approach defining several measures on the basis of attributes obtained from design artifacts [3]. They reported on a case study, and two replications, conducted with students of an advanced university class, that highlighted the impact of some attributes, obtained from artifacts designed with W2000, on the total effort required for designing Web applications. It is worth noting that, in our empirical study, Web applications were not automatically obtained from design documents, thus our focus was on the effort to develop Web applications.

Abrahão and Pastor proposed the OOmFPWeb method [1] which maps the Function Points concepts into the primitives used in the conceptual modeling phase of OOWS, a method for producing software for the Web. In a recent work, an initial validation of the proposed size measure was described [2].

## 4 Conclusions

In this paper we have investigated the use of Web-COBRA proposed in [21] for estimating Web application development effort. In particular, we have replicated a previous case study [21], by applying Web-COBRA in combination with the Web Objects measure and exploiting data from 15 industrial Web applications developed by an Italian software company. The empirical analysis has not only confirmed the effectiveness of Web-COBRA and Web Objects in estimating development effort, but has also shown their ability to scale, dealing with bigger applications.

In the future, we intend to further assess Web-COBRA by considering a different contest. Moreover, we plan to apply Web-COBRA using COSMIC as size measure [7]. Indeed, Web-COBRA's authors suggested to employ COSMIC or Web Objects, however in [21] only Web Objects approach was used.

## References

[1] Abrahão, S.M., Pastor, O.: Measuring the functional size of Web applications. International Journal of Web Engineering and Technology 1(1), 5–16 (2003)

[2] Abrahão, S.M., Pastor, O., Poels, G.: Evaluating a Functional Size Measurement Method for Web Applications: An Empirical Analysis. In: Proceedings of International Software Metrics Symposium (METRICS 2004), pp. 358–369 (2004)

[3] Baresi, L., Morasca, S.: Three Empirical Studies on Estimating the Design Effort of Web Applications. Transaction On Software Engineering and Methodology 16(4) (2007)

[4] Briand, L., El Emam, K., Bomarius, F.: COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking, and Risk Assessment. In: Proceedings of the International Conference on Software Engineering (ICSE 1998), pp. 390–399 (1998)

[5] Cohen, J.: Statistical Power Analysis for the Behavioral Science. Lawrence Erlbaum Hillsdale, New Jersey (1988)

[6] Conte, D., Dunsmore, H.E., Shen, V.Y.: Software engineering metrics and models. The Benjamin/Cummings Publishing Company, Inc. (1986)

[7] COSMIC. Web site (2007), http://www.cosmicon.com

[8] 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 International Conference on Web Engineering (ICWE 2006), pp. 9–16 (2006)

[9] 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), 93–120 (2006)

[10] IFPUG, Function point counting practices manual, release 4.2.1 (2004)

[11] Kampenes, V., Dybå, T., Hannay, J.E., Sjøberg, D.I.K.: A Systematic Review of Effect Size in Software Engineering Experiments. Information and Software Technology 4(11-12), 1073–1086 (2007)

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

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

[14] Linstone, H.A., Turoff, M.: The Delphi Method: Techniques and Applications. Addison-Wesley Publishing Co. Inc. (1975)

[15] 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)

[16] 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(2), 163–196 (2003)

[17] Mendes, E., Counsell, S., Mosley, N., Triggs, C., Watson, I.: A Comparison of Development Effort Estimation Techniques for Web Hypermedia Applications. In: Proceedings of International Software Metrics Symposium (METRICS 2002), pp. 131–140 (2002)

[18] 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 (METRICS 2004), pp. 348–357 (2004)

[19] Mendes, E., Mosley, N.: Bayesian Network Models for Web Effort Prediction: A Comparative Study. IEEE Transactions on Software Engineering (August 1, 2008), http://doi.ieeecomputersociety.org/10.1109/TSE.2008.64

[20] Reifer, D.: Web-Development: Estimating Quick-Time-to-Market Software. IEEE Software 17(8), 57–64 (2000)

[21] Ruhe, M., Jeffery, R., Wieczorek, I.: Cost estimation for Web applications. In: Proceedings of the International Conference on Software Engineering (ICSE 2003), pp. 285–294 (2003)