Comparing Size Measures for Predicting Web Application Development Effort: A Case Study

Sergio Di Martino, Filomena Ferrucci,
Carmine Gravino
Università di Salerno
Via Ponte Don Melillo, I-84084
Fisciano (SA) Italy, 0039 089963374
{sdimartino,fferrucci,gravino}@unisa.it

Emilia Mendes
The University of Auckland
Private Bag 92019
Auckland, New Zealand
0064 9 3737599 ext. 86137
emilia@cs.auckland.ac.nz

Abstract

Size represents one of the most important attribute of software products used to predict software development effort. In the past nine years, several measures have been proposed to estimate the size of Web applications, and it is important to determine which one is most effective to predict Web development effort. To this aim in this paper we report on an empirical analysis where, using data from 15 Web projects developed by a software company, we compare four sets of size measures, using two prediction techniques, namely Forward Stepwise Regression (SWR) and Case-Based Reasoning (CBR). All the measures provided good predictions in terms of MMRE, MdMRE, and Pred(0.25) statistics, for both SWR and CBR. Moreover, when using SWR, Length measures and Web Objects gave significant better results than Functional measures, however presented similar results to the Tukutuku measures. As for CBR, results did not show any significant differences amongst the four sets of size measures.

1. Introduction

Estimation of development effort is an important management activity for planning and monitoring software development projects. Accurate early estimations are crucial to allocate resources adequately and to deliver products on time and within budget. Clearly, this is a critical activity for the competitiveness of a software company.

In the context of Software- and Web-engineering, many techniques have been applied to estimate the effort necessary to develop a new project, such as (Stepwise) Linear Regression and Case-Based Reasoning *(*or Analogy-Based Estimation) (see e.g., [3],[4],[9],[10],[12],[16],[17],[19],[20],[21],[24],[25]). These techniques use data from past projects, characterized by attributes that are related to effort, and the actual effort to develop the projects, to estimate effort for a new project under development (see e.g., [3],[4],[6],[10]).

The techniques used and the characteristics of the datasets can play a key role in the accuracy of the predictions that are obtained [26]. Several Web size measures have been proposed to date to be used as Web effort predictors (see e.g., [2],[9],[10],[14],[16], [17],[21],[23],[24],[25]). However, there is not yet a widely accepted Web size measure, since several empirical investigations would have been necessary to compare and validate/confirm the usefulness of a measure as effort predictor. One of the main drawbacks to carry out such empirical studies is the lack of publicly available industrial datasets. To the best of our knowledge, to date only three papers have reported on case studies aimed at assessing the effectiveness of different size measures for Web cost estimation, by providing direct comparisons [10],[19],[25]. Mendes *et al.* [19] compared, using multivariate regression techniques, Web-based *Length* size measures with a conventional function points size measure using the *COSMIC-FFP* method [8]. Their empirical results revealed that "none of the obtained models produced reasonable accurate estimates of the effort", and "the models did not produce significantly different residual values" [8].

Ruhe *et al.* [25] compared, using multivariate regression, *Web Objects* with *Function Points* (*FPs*).

0-7695-2886-4/07 $20.00 © 2007 IEEE
DOI 10.1109/ESEM.2007.20

FPs is a de facto standard used to estimate the size of traditional business systems and to indirectly predict their effort, cost, and duration [1]. *Web Objects* were introduced by Reifer as an extension of *FPs*, specifically designed for Web applications [22][23]. The results of the empirical analysis revealed that the model based on *Web Objects* presented significantly better prediction accuracy.

Costagliola *et al.* compared Web-based *Length* and *Functional* measures using both Stepwise Linear Regression and Case Base Reasoning [10]. Their empirical results revealed that *Length* measures provided better estimates when using *Case Base Reasoning*, while *Functional* measures *provided better estimates* when using *Stepwise Regression*. However, their analysis suggested that there were no significant differences in the estimations and the residuals obtained with *Length* measures (using *Case Base Reasoning)* and *Functional* measures (by means of *Stepwise Regression)*.

The need for further empirical investigations is the motivation for this paper. We compared, using data from 15 Web projects developed by an Italian software company, the following size measures: *Web Objects* [22], the *Length* and *Functional* measures used by Costagliola et al. in [10], the *Tukutuku* measures proposed by Mendes *et al.* [17].

The effort estimation techniques used to compare the size measures are Forward Manual Stepwise Regression (SWR), as proposed in [14],[16], and Case-Based Reasoning (CBR).

The remainder of the paper is organized as follows. Section 2 describes the dataset used for the case study. The results of the empirical analysis obtained using SWR and CBR are presented in Section 3. A discussion of the results is provided in Section 4, and conclusions and comments on future work are given in Section 5.

2. Dataset and Size Measures

We have used in our analysis data on 15 Web applications, developed by a medium-size Italian software company. This company's core business is the development of enterprise information systems, mainly for local and central government. Among other clients, there are educational structures, health organizations, research centers, industries, and other public institutions. It is specialized in the design, development, and management of solutions for Web portals, enterprise intranet/extranet applications (such as content-ware, e-commerce, and work-flow managers), and Geographical Information Systems. The company has about fifty employees, and is certified ISO 9001 for software development. Its turnover in the year 2003 was about 5M €.

The Web projects used in our empirical study represent several domains, such as: e-government, e-banking, Web portals, and intranet applications. They were developed using several technologies, such as J2EE, ASP and ASP.NET. Oracle was the database system used by most projects, and SQL Server, Access and MySQL were also employed in some projects.

With regard to the validity of our study, for each set of size measures used in our empirical study (*Tukutuku* [17], *Web Objects* [22], *Length* and *Functional* [10] variables) data have been obtained both from analysis and design documents, and using questionnaires filled out by the project managers of the software company. As for the effort collection, the software company employed a timesheet to keep track of this information. Each team member entered daily the information about his/her development effort, and on a weekly basis project managers stored the sum of the team effort.

As for the measures, authors defined a template to be filled in by the project managers, in order to collect all the significant information to calculate the values of the measures. All the project managers were trained on the questionnaires, to correctly provide the required information. Finally, in order to cross-check the provided information, one of the authors analyzed the filled templates and the analysis and design documents related to the projects.

The following sub-sections describe the size measures used in this study.

2.1 Tukutuku variables

Data about the 15 Web projects have been recently included in the Tukutuku database [17], part of the Tukutuku project[1], which aims to collect data about Web applications, to be used to develop Web effort estimation models and to benchmark productivity across and within Web Companies.

Each Web project was characterized by 25 variables, related to the application and its development process. These size measures and cost drivers have been defined from the results of a survey investigation [18], using data from 133 on-line Web forms aimed at giving quotes on Web development projects. In addition, these measures and cost drivers have also been confirmed by an established Web company and a second survey involving 33 Web

[1] http://www.cs.auckland.ac.nz

companies in New Zealand. Table 1 shows the size measures only, as these were the ones used in our investigation.

We excluded from our analysis some variables on the basis of the following criteria:

- More than 40% of instances of a variable missing.
- The variable did not measure size per se.

Table 1 - Variables for the Tukutuku database

Variable Name	Scale	Description
TotEff	Ratio	Actual total effort used to develop the Web application.
TotWP	Ratio	Number of new and reused Web pages.
NewWP	Ratio	Number of new Web pages.
TotImg	Ratio	Number of new and reused images.
NewImg	Ratio	Number of new images created.
Fots	Ratio	Number of features reused without any adaptation.
HFotsA	Ratio	Number of reused high-effort features/functions adapted.
Hnew	Ratio	Number of new high-effort features/functions.
TotHigh	Ratio	Number of high-effort features/functions
FotsA	Ratio	Number of reused low-effort features adapted.
New	Ratio	Number of new low-effort features/functions.
TotNHigh	Ratio	Number of low-effort features/functions

Table 2 shows summary statistics for the Tukutuku variables in Table 1.

Table 2 – Summary statistics for the Tukutuku measures

	Mean	Median	Std. Dev.	Min.	Max.
TotWP	84.13	74	40.56	31	161
NewWP	79.6	70	41.43	31	161
TotImg	172	96	193.89	64	829
NewImg	18.00	0	28.16	0	92
Fots	4.87	5	4.76	0	15
Hnew	15.60	14	6.09	7	27
TotHigh	15.73	16	6.08	7	27
New	6.60	5	3.40	3	13
TotNHigh	7.07	6	3.35	3	14
TotEff	2,677.87	2,792	827.11	1,176	3,712

2.2 Web Objects

Web Objects [23] are an extension of *Function Points* (*FPs*), a method introduced by Albrecht to estimate software size of business systems early on in the development life cycle [1][11]. *Web Objects* extends *FPs* by introducing four Web-related components (*multimedia files*, *Web building blocks*, *scripts* and *links*), to be used together with the five traditional function types of *FPs* (*external input, external output, external inquiry, internal logical file, and external interface file*) to compute the functional size of a Web application [23].

Reifer devised such list of components based on the opinion of experts and by analyzing 64 completed Web projects in five application domains[2]. A description of these predictors is reported in Table 3.

Table 3. Web Objects Components

Variable Name	Scale	Description
Internal Logical Files (ILF)	Ratio	logical, persistent entities maintained by the Web application to store information of interest.
External Interface Files (EIF)	Ratio	logical, persistent entities that are referenced by the Web application, but are maintained by another software application
External Inputs (EI)	Ratio	logical, elementary business processes that cross into the application boundary to maintain the data on an Internal Logical File, access a Multi-Media File, invoke a Script, access a Link or ensure compliance with user requirements
External Outputs (EO)	Ratio	logical, elementary business processes that result in data leaving the application boundary to meet a user requirements (e.g., reports, screens).
External Queries (EQ)	Ratio	logical, elementary business processes that consist of a data "trigger" followed by a retrieval of data that leaves the application boundary (e.g., browsing of data).
Multi-Media Files (MMF)	Ratio	physical, persistent entities used by the Web application to generate output in multi-media format.
Web Building Blocks (WBB)	Ratio	logical persistent entities used to build the Web applications and automate their functionality
Scripts (Scr)	Ratio	logical, persistent entities used by the Web application to link internal files and building blocks together in predefined patterns
Links (Lin)	Ratio	logical, persistent entities maintained by the Web application to find links of interest to external applications

The functional size of a Web application, in number of *Web Objects*, is determined by measuring the nine components using as input design documents. As suggested in [23], first the instances of the components are counted, and a complexity (low, average, high) is associated with them. Then, by using a calculation worksheet, a weight is associated to each counted instance of the Web components. Thus, the number of *Web Objects* is given by summing up all these weights.

[2] Note that despite Reifer's claim the results from using these 64 projects were never made publicly available.

Table 4 reports the summary statistics of the size expressed in *Web Objects* (denoted as *WO*) for the 15 Web projects of our case study.

Table 4 – Summary statistics for the Web Objects measure

	obs	Mean	Median	Std. Dev.	Min.	Max.
WO	15	1,474.867	1389	543.417	465	2,258

2.3 Length and Functional measures

The other two size measures taken into account in our case study are those previously used by Costagliola *et al.* [10], organized in two sets of variables denoted as *Length* and *Functional* measures. In particular, *Length* measures (see Table 5), were derived from both previous research (e.g. [19],[20]) and interviews with the company's project managers. As for the *Functional* measures, Costagliola *et al.* used the nine components (see Table 3) that are part of *Web Objects*.

Table 5: Length measures [10]

Variable	Scale	Description
Wpa	Ratio	Number of Web pages
Me	Ratio	Number of multimedia elements
N_Me	Ratio	Number of new multimedia elements
CSAPP	Ratio	Number of Client side Scripts and Applications
SSApp	Ratio	Number of Server side Scripts and Applications
IL	Ratio	Number of Internal Links
EL	Ratio	Number of External References

Table 6 contains the summary statistics of both the *Length* and the *Functional* measures for the 15 Web projects.

Table 6: Descriptive statistics of the Length and Functional measures

	Mean	Median	Std. Dev.	Min.	Max.
Wpa	17	11	12.317	2	46
Me	104.133	93	43.533	54	223
N_Me	82.533	63	56.599	20	223
CSAPP	26.933	36	16.918	5	55
SSApp	80,4	65	55.414	2	209
EL	4.933	8	3.770	0	8
IL	279.133	235	145.322	124	592
EI	24.533	18	18.302	2	59
EO	20.2	19	11.965	5	41
EQ	40.267	34	27.044	7	102
ILF	2.733	3	2.604	0	7
EIF	5.667	4	4.624	1	15
WBB	27.867	31	14.252	12	53
MMF	100	94	49.558	15	225
Scr	139.4	130	62.810	56	260
Lin	366.8	172	172.825	172	655

3 Empirical Analyses and Results

The following sub-sections present the empirical analyses and the results of our investigation using forward Stepwise Regression (SWR) and Case-Based Reasoning (CBR). Except for CBR, all results presented here were obtained using the statistical software SPSS 13.0 for Windows. Finally, all the statistical significance tests used $\alpha = 0.05$.

3.1 Obtaining effort estimates using forward stepwise regression

Stepwise Regression [15] is a statistical technique whereby a prediction model (Equation) is built, and represents the relationship between independent (e.g. number of Web pages) and dependent variables (e.g. total Effort). This technique builds the model by adding, at each stage, the independent variable with the highest association to the dependent variable, taking into account all variables currently in the model. It aims to find the set of independent variables (predictors) that best explains the variation in the dependent variable (response). In particular, we applied a manual stepwise regression using the technique proposed by Kitchenham [12]. Basically the idea is to use this technique to select the important independent variables, and then to use linear regression to obtain the final model.

In our study we employed the variables shown in Tables 2, 4 and 6 with manual stepwise regression in order to select the most important size measures. Once selected they were the ones used for cross-validation, i.e. we did not perform a separate manual stepwise regression for each cross-validation step; we simply performed a regression using the variables previously selected using the manual stepwise procedure (see Equations 1, 3 and 5).

Whenever variables were highly skewed they were transformed before being used in the forward stepwise procedure. This was done in order to comply with the assumptions underlying stepwise regression [15] (i.e. residuals should be independent and normally distributed; relationship between dependent and independent variables should be linear). The transformation employed was to take the natural log (Ln), which makes larger values smaller and brings the data values closer to each other [15]. A new variable containing the transformed values was created for each original variable that needed to be transformed. All new variables are identified as *Lvarname*, e.g. *LTotEff* represents the transformed variable *TotEff*. In addition, whenever a variable needed to be transformed but had zero values, the natural logarithmic transformation was applied to the variable's value after adding 1.

To verify the stability of each effort model built using forward stepwise regression, the following steps were employed [14]:

- Use of a residual plot showing residuals vs. fitted values to investigate if the residuals are randomly and normally distributed.
- Calculate Cook's distance values [7] for all projects to identify influential data points. Any projects with distances higher than 3 × (4/*n*), where *n* represents the total number of projects, are immediately removed from the data analysis [15]. Those with distances higher than 4/n but smaller than (3 × (4/n)) are removed in order to test the model stability, by observing the effect of their removal on the model. If the model coefficients remain stable and the adjusted R^2 (goodness of fit) improves, the highly influential projects are retained in the data analysis.

3.1.1 Tukutuku measures

The best fitting model obtained by applying SWR to the Tukutuku variables (see Table 1) is described in Table 7. Observe that SWR identified *TotHigh* as the preeminent effort predictor, thus suggesting that most of the development effort is devoted to implementing server-side functions and features. The model's adjusted R^2 was 0.714, thus it explains 71.4% of the variation in *TotEff*.

The Equation as read from the final model's output is:

$$TotEff = 842.720 + 116.641 TotHigh \quad (1)$$

Table 7 – The Best Fitting Model to calculate TotEff using the Tukutuku variables

	Coefficient	Std. Error	t	p>|t|
(constant)	842.720	326.782	2.579	.023
TotHigh	116.641	19.460	5.994	.000

The P-P plot (Probability plot) and the residual plot are presented in Figure 1(a) and Figure 1(b) respectively. P-P Plots are normally employed to verify if the distribution of a variable matches a given distribution, in which case data points gather around a straight line. The distribution which has been checked here is the normal distribution, and Figure 1(a) suggests that the residuals are normally distributed.

The residual plot for the 15 projects showed that one project that seemed to have a large residual. This trend was also confirmed using Cook's distance, where these projects had their Cook's distances above the cut-off point (4/15). To check the model's stability, a new model was built without this project, giving an adjusted R^2 of 0.810, which is greater than that for the previous model. In the new model the independent variables remained significant and the coefficients had very similar values to those in the previous model, indicating that the high influence data point did not need to be permanently removed from further analysis.

3.1.2 Web Objects

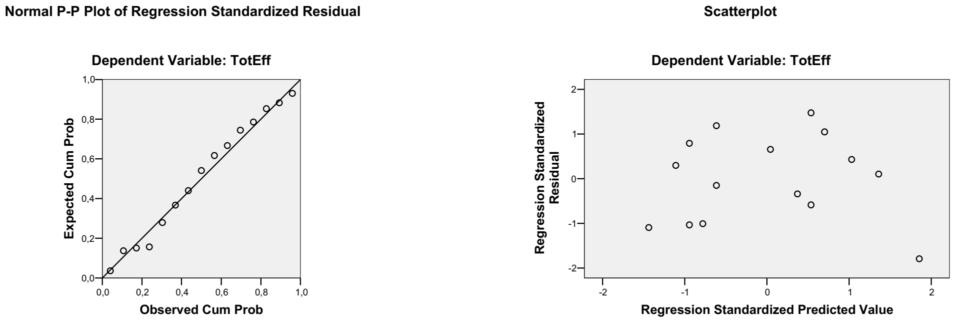
As for the *Web Objects* measure, since we had only one variable, we applied a simple linear regression that provided the model described in . The adjusted R^2 was 0.647, thus the *Web Objects* measure explains 64.7% of the variation in *TotEff*.

Table 8 - The Model built using Web Objects

	Coefficient	Std. Error	t	p>|t|
(constant)	851.912	375.814	2.267	.041
WO	1.246	0.241	5.162	.000

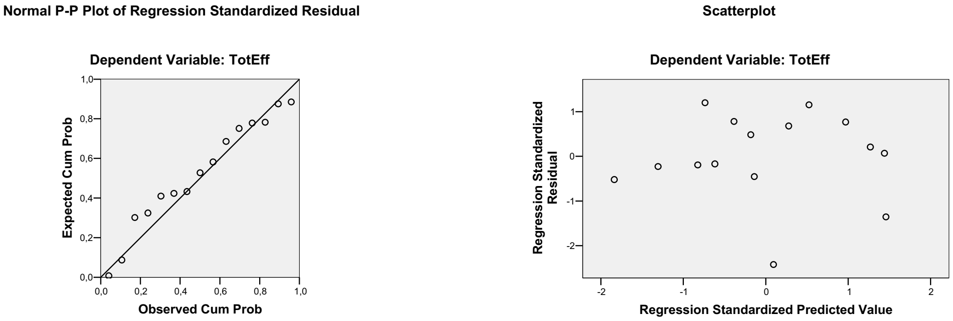
The Equation as read from the final model's output is:

$$TotEff = 851.912 + 1.246 WO \quad (2)$$



(a) **(b)**

Figure 1 – P-P plot (a) and Residual plot (b) for the model obtained using Tukutuku measures



(a) (b)

Figure 2 – P-P plot (a) and Residual plot (b) for the model obtained using Web Objects

The P-P plot and the residual plot are presented in Figure 2(a) and Figure 2(b) respectively. Figure 2(a) suggests that the residuals are normally distributed. The residual plot showed two projects that seemed to have a large residual. This trend was also confirmed using Cook's distance. However, the model turned out to be stable by removing these two projects (giving an adjusted R^2 of 0.884 and similar coefficients). Thus, these data points were not permanently removed from further analysis.

3.1.3 Length measures

The best fitting model obtained by applying SWR to the Length measures is described in Table 9. In this case, SWR has identified three factors as main effort predictors: the number of server-side applications (*LSSApp*), the number of Internal Links to other components (*LIL*), and the number of Multimedia Elements (*LME*). Model's adjusted R^2 is 0.841, thus these variables explain 84.1% of the variation in *TotEff.* As we can see, this adjusted R^2 value is greater than those obtained with *Tukutuku* and *Web Objects* measures; however this model selected three variables, which increases the probability of obtaining a greater adjusted R^2.

The Equation as read from the final model's output is:

$$LTotEff = 4.358 + 0.508LSSApp + 0.192LIL + 0.241LMe \quad (3)$$

which, when transformed back to the raw data scale, gives the Equation:

$$TotEff = 78 \times SSApp^{0.508} \times IL^{0.192} \times Me^{0.241} \quad (4)$$

Table 9 - The Model built using Length Measures

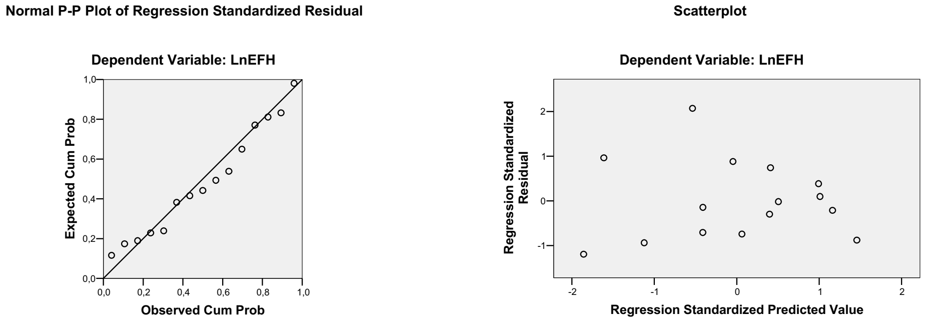
	Coefficient	Std. Error	t	p>|t|
(constant)	4.358	0.511	8.523	0.000
LSSApp	0.508	0.055	9.235	0.000
LCSApp	0.192	0.036	5.289	0.000
LMe	0.241	0.093	2.589	0.025

The P-P plot (see Figure 3(a)) suggests that the residuals are normally distributed. The residual plot of Figure 3(b) revealed that one project presented a large residual. For that project the Cook's distance was greater than 4/15. To check the model's stability, a new model was generated without this project. In the new model the independent variables remained significant, the adjusted R^2 improved a little, and the coefficients present similar values to those in the previous model. Thus, the data point was not permanently removed from further analysis.

It is worth noting that the study reported in [10] was not carried out employing Manual SWR but the SWR procedure as supported in the SPSS tool. As a matter of fact, in that case different variables were selected as predictors, namely *Wpa* and *Me*.

3.1.4 Functional measures

The best fitting model obtained by applying SWR to the Functional measures is described in Table 10. In this case, SWR identified *External Inputs* (basically the number of Web forms) as the main factor affecting the development effort. The model's adjusted R^2 was 0.513, thus it explains 51.3% of the variation in *TotEff.*



(a) **(b)**

Figure 3 – P-P plot (a) and Residual plot (b) for the model obtained using Length measures

Table 10 – Best Fitting Model to calculate TotEff

	Coefficient	Std. Error	t	p>|t|
(constant)	7.492	0.108	69.206	.000
EI	0.014	0.004	3.790	.002

The Equation as read from the final model's output is:

$$LTotEff = 7.492 + 0.014EI \qquad (5)$$

which, when transformed back to the raw data scale, gives the Equation:

$$TotEff = 1794 \times e^{0.048EI} \qquad (6)$$

The P-P plot presented in Figure 4(a) suggests that the residuals are normally distributed. Although the residual plot (see Figure 4(b)) showed one project with a large residual and this trend was also confirmed using Cook's distance, the analysis of the stability of the model suggested there was no need to remove the high influence data point from further analysis.

As for the study reported in [10], the SPSS tool selected as predictors the variables *EI* and *Lin*.

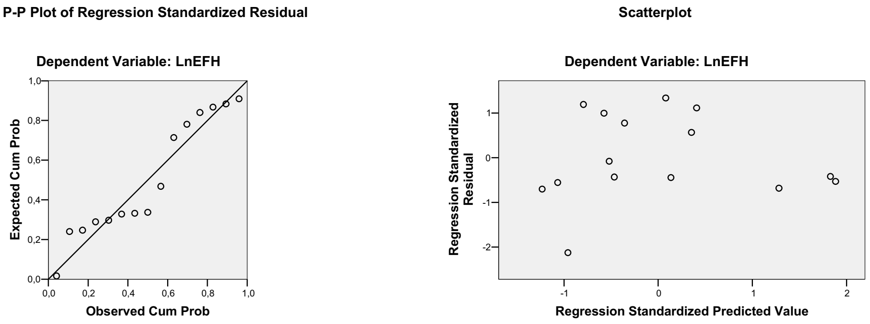
3.2 Obtaining effort estimates using case-based reasoning

Case-Based Reasoning (CBR) is a branch of Artificial Intelligence where knowledge of similar past cases is used to solve new cases [26]. Within the context of our investigation, the idea behind the use of the CBR technique is to predict the effort of a new project by considering similar projects previously developed. In particular, the completed projects are characterized in terms of a set of *p* features and form the *case base*. The new project is also characterized in terms of the same *p* features and it is referred as the *target case*. Then, the similarity between the target case and the other cases in the *p*-dimensional feature space is measured, and the most similar cases are used, possibly with adaptations to obtain a prediction for the target case. To apply the method, we have to select the relevant project features, the appropriate similarity function, the number of analogies to choose the similar projects to consider for estimation, and the analogy adaptation strategy for generating the estimation. The selection of the similarity function and the number of analogies are crucial decisions. The similarity measure used in this study is the Euclidean distance as this has been the measure used in the literature with the best results [21]. In addition, all the project attributes considered by the similarity function had equal influence upon the selection of the most similar project(s).

In particular, to apply CBR, we have employed the ANGEL tool [26] by using as set of features: *Web Objects* (Table 3), *Tukutuku* (Table 4), *Length,* and *Functional* (Table 6) measures. We employed ANGEL's *Feature Subset Selection* (FSS), which determines the optimum subset of features that yield the most accurate estimation. ANGEL applies FSS using an exhaustive search and a jack knife approach [26], which in our case was computationally tractable since the number of features was small. Estimates were based on the average effort of the two most similar projects in the case base, with no different weights for attributes or adaptation of the estimated effort. Results are described in the next Section.

4. Comparison of the prediction accuracy

The accuracy of the effort estimates was assessed by applying a *leave-one-out cross-validation*. Cross-validation is the splitting of a dataset into training and validation sets. Training sets are used to build models and validation sets are used to validate models.



(a) **(b)**

Figure 4 – P-P plot (a) and Residual (b) for the model obtained using Functional measures

A leave-one-out cross-validation means that the original dataset is divided into *n* different subsets (*n* is the size of the original dataset) of training and validation sets, where each validation set has one project. The equivalent for CBR is to use the dataset as a case base, after removing one project, and then to estimate effort for the project that has been removed. This step is iterated *n* times, each time removing a different project.

To assess the accuracy of the derived effort prediction models, we have employed de facto standard accuracy measures, such as the mean Magnitude of Relative Error (MMRE), median MRE (MdMRE), and Prediction at 25% (Pred(25)) [6].

Pred(*n*) measures the percentage of estimates that are within *n*% of the actual values, and *n* is usually set at 25%. MRE is the basis for calculating MMRE and MdMRE, and defined as:

$$\text{MRE} = \frac{|e - \hat{e}|}{e} \tag{8}$$

where e represents actual effort and ê estimated effort. The difference between MMRE and MdMRE is that the former is sensitive to predictions containing extreme MRE values.

In the following sub-section we compare the prediction accuracies taking into account first, the summary statistics and second, the boxplots of absolute residuals, where residuals are calculated as (actual effort – estimated effort).

As suggested by Mendes and Kitchenham [14],[16], we also analyzed MMRE, MdMRE, and Pred(0.25) obtained by considering at each step of the leave-one-out cross validation the mean of effort (i.e., MeanEffort) and the median of effort (i.e., MedianEffort) as predicted value. The aim is to have a benchmark to assess whether the estimates obtained with SWR and CBR are significantly better than estimates based on the mean or median effort.

4.1 Comparison using summary statistics

The values of MMRE, MdMRE, and Pred(0.25) obtained using SWR and CBR on the considered sets of size measures are shown in Table 11. SWR-Tuk (CBR-Tuk) denotes the application of SWR (CBR) using the *Tukutuku* measures; SWR-WO (CBR-WO) denotes the application of SWR (CBR) using the *Web Objects measure*; SWR-Leng (CBR-Leng) denotes the application of SWR (CBR) using the *Length* measures; SWR-Funct (CBR- Funct) denotes the application of SWR (CBR) using the *Functional* measures.

Table 11: MMRE, MdMRE, and Pred(0.25)

Technique-Measure	MMRE	MdMRE	Pred(0.25)
SWR-Tuk	0.18	0.14	0.73
SWR-WO	0.17	0.11	0.80
SWR-Leng	0.12	0.11	0.87
SWR-Funct	0.23	0.21	0.73
CBR- Tuk	0.16	0.12	0.87
CBR- WO	0.21	0.11	0.80
CBR- Leng	0.18	0.12	0.87
CBR- Funct	0.14	0.11	0.93
MeanEffort	0.34	0.27	0.47
MedianEffort	0.33	0.24	0.60

Overall, both SWR and CBR predictions were good if we assume as a reasonable threshold Conte *et al.*'s suggestion that good predictions should present a MMRE and MdMRE not greater than 25% and Pred(25) greater or equal to 75% [6]. However, predictions obtained using CBR were superior to those obtained using SWR. Overall, if we employ as basis MMRE, MdMRE and Pred(25), the best results for CBR and SWR were obtained using *Functional* measures and *Length* measures, respectively.

The results obtained with CBR for *Length* and *Functional* measures are better than those in [10], where ANGEL's FSS was not used, thus showing that using FSS does improve estimations.

We also used a non-parametric test – Kendall's W test[3], and the absolute residuals, to check whether there were statistically significant differences between the four different residuals samples. The residuals obtained using SWR showed that *Length* measures and *Web Objects* presented significantly superior predictions than *Functional* measures; however all presented similar predictions to the *Tukutuku* size measures. With regard to the residuals obtained using CBR, our results did not show any significant differences amongst the four size measures. When comparing residuals between SWR and CBR, Kendall's W test revealed that the *Functionality* measures using CBR presented significantly superior predictions than these same measures using SWR.

Finally, Kendall's W Test showed that only the estimations obtained using *Length* measures with SWR were significantly better than those obtained using *MeanEffort* and *MedianEffort;* whereas the estimations obtained using the *Tukutuku* measures with CBR were significantly better than those obtained with *MeanEffort*.

4.2. Comparison using Boxplots

To compare the accuracy between the obtained estimates we also used the absolute residuals. Their boxplots are presented in Figure 5, and confirm the results obtained with MMRE, MdMRE, and Pred(0.25) statistics. In particular, they show that the spread of the distributions for CBR are slightly wider than those for SWR; however most SWR boxplots present outliers. If we look at the medians they indicate that SWR-Funct presents the largest residuals, followed by SWR-Tuk, CBR-WO and SWR-WO. According to the boxplots, the best predictions were obtained for SWR-Leng and CBR-Funct, followed by CBR-Tuk. The best result using SWR has been obtained for *Length* measures, also confirmed by MMRE, MdMRE, and Pred(0.25) statistics. As for CBR the best results has been obtained with *Functional* measures, also confirmed by MMRE, MdMRE, and Pred(0.25) statistics.

5. Final remarks

We have provided the results of an empirical investigation, based on an industrial dataset, meant to compare size measures for estimation of Web application development effort. In particular, we have focused on four sets of size measures: *Tukutuku* measures [14],[16], *Web Objects* [22], *Length* and *Functional* measures [9][10]. To obtain the estimates we have employed Forward Stepwise Regression (SWR) and Case-Based Reasoning (CBR), which have been widely adopted in the literature (see e.g., [10],[16],[19],[24]).

The empirical results showed that all the measures provided good predictions in terms of MMRE, MdMRE, and Pred(0.25) statistics, for both SWR and CBR. Moreover, when using SWR*,* *Length* measures and *Web Objects* gave significant better results than *Functional* measures, however they presented similar results to the *Tukutuku* measures. As for CBR, results did not show any significant differences amongst the size measures. Finally, CBR presented significant better results than SWR when using *Functional* measures.

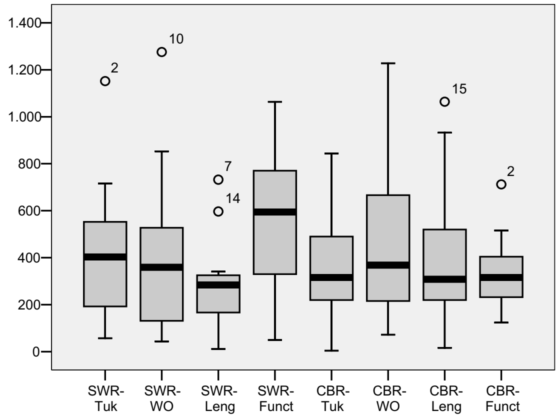


Figure 5 – The boxplots of absolute residuals

This study has largely confirmed the results of previous work, revealing that both SWR and CBR can be profitably exploited to predict Web application development effort [10][21]. However, some differences have arisen with respect to the results of [10] and [21]. In particular, in contrast to [10], in the current study Stepwise Regression has provided positive results also with Length measures. This might be motivated by the application of manual selection of the variables to be used in our models. About [21], in that study the authors showed that SWR provided statistically significantly superior predictions than other techniques when using Length measures, but this is not confirmed in our study.

What these results suggest to practitioners is that Web companies that develop projects with similar characteristics to those used in our study can use

[3] We used a non parametric test since the residuals were not normally distributed in two cases (SWR-2, CBR-2).

Length size measures for Web cost estimation if they employ SWR to obtain effort estimates; and *Tukutuku* measures if they employ CBR to obtain effort estimates. However, if a company does not use any formal technique to obtain estimates either *Length* measures, *Web Objects* or the *Tukutuku* measures are good choices.

As future work, we plan to conduct further research on larger datasets of Web projects. Moreover, other parameters and/or adaptation strategies should be employed to further investigate the CBR technique.

6. References

[1] A.J. Albrecht, "Measuring Application Development Productivity," in *Procs. Joint SHARE/GUIDE/IBM Application Development Symposium*, 1979, pp. 83-92.

[2] L. Baresi, S. Morasca, P. Paolini, "Estimating the Design Effort of Web Applications," in *Procs. 9th Intl. Software Metrics Symposium* (METRICS'03), 2003, pp. 62-72.

[3] L.C. Briand, K. El-Emam, K. Maxwell, D. Surmann, I. Wieczorek, "An assessment and comparison of common cost estimation models", in *Procs. 21st Intl. Conference on Software Engineering* (ICSE'99), 1999, pp. 313-322

[4] L.C. Briand, T. Langley, I. Wieczorek, "A replicated assessment of common software cost estimation techniques", *in Procs. 22nd Intl. Conference on Software Engineering* (ICSE'00), 2000, pp.377-386.

[5] S. P. Christodoulou, P. A. Zafiris, T. S. Papatheodorou, "WWW2000: The Developer's view and a practitioner's approach to Web Engineering", in *Procs. Second ICSE Workshop on Web Engineering*, 2000, pp. 75-92.

[6] S.D. Conte, H.E. Dunsmore, V.Y. Shen, "Software Engineering Metrics and Models", Benjamin-Cummins, 1986.

[7] R.D. Cook, "Detection of influential observations in linear regression, *Technometrics*, 19, 1977, pp. 15-18.

[8] COSMIC: *COSMIC-FFP* Measurement manual, version 2.2, http://www.cosmicon.com, 2003.

[9] G. Costagliola, S. Di Martino, F. Ferrucci, C. Gravino, G. Vitiello, G. Tortora, "A COSMIC-FFP Approach to Predict Web Application Development Effort", *Journal of Web Engineering*, 5(2), 2006, pp. 93-120.

[10] G. Costagliola, S. Di Martino, F. Ferrucci, C. Gravino, G. Tortora, G. Vitiello, "Effort estimation modeling techniques: a case study for web applications", in *Procs. Intl. Conference on Web Engineering* (ICWE'06), 2006, pp. 9-16.

[11] International Function Point Users Group: "Function Point Counting Practices Manual," Release 4.1.1, 2001.

[12] B. A. Kitchenham, "A Procedure for Analyzing Unbalanced Datasets", *IEEE TSE*, 24(4), 1998, 278-301.

[13] B. A. Kitchenham, E. Mendes, "Software Productivity Measurement Using Multiple Size Measures"*, IEEE TSE*, 30(12), 2005, pp. 1023-1035.

[14] B. A. Kitchenham, E. Mendes, "A Comparison of Cross-company and Single-company Effort Estimation Models for Web Applications", in *Procs. EASE 2004*, 2004, pp. 47-55.

[15] K. Maxwell, "Applied Statistics for Software Managers". Software Quality Institute Series, Prentice Hall, 2002.

[16] E. Mendes, B.A. Kitchenham, "Further Comparison of Cross-Company and Within Company Effort Estimation Models for Web Applications", *Procs. Intl. Software Metrics Symposium* (METRICS'04), 2004, pp. 348-357.

[17] E. Mendes, N. Mosley, S. Counsell, "Investigating Early Web Size Measures for Web Cost Estimation", in *Procs. EASE 2003*, pp. 1-22.

[18] E. Mendes, N. Mosley, S. Counsell, "Investigating Web Size Metrics for Early Web Cost Estimation", *Journal of Systems and Software*, 77(2), 2005, pp. 157-172.

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

[20] E. Mendes, S. Counsell, N. Mosley, "Web Metrics – Estimating Design and Authoring Effort", *IEEE Multimedia*, 2001, pp. 50-57.

[21] E. Mendes, S. Counsell, N. Mosley, C. Triggs, I. Watson, "A Comparative Study of Cost Estimation Models for Web Hypermedia Applications", *Empirical Software Engineering* 8(2), 2003, pp. 163-196.

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

[23] D. Reifer, "Web Objects Counting Conventions", *Reifer Consultants*, Mar. 2001. Available at: http://www.reifer.com/download.html.

[24] M. Ruhe, R. Jeffery, I. Wieczorek, "Cost estimation for web applications", in *Procs. Intl. Conference on Software Engineering* (ICSE'03), 2003, pp. 285–294.

[25] M. Ruhe, R. Jeffery, I. Wieczorek, "Using Web Objects for Estimating Software Development Effort for Web Applications", in *Procs. Intl. Software Metrics Symposium* (METRICS'03), 2003, pp. 30-37.

[26] M.J.Shepperd,, G. Kadoda, Using Simulation to Evaluate Prediction Techniques, in *Procs. IEEE Intl. Software Metrics Symposium* (METRICS'01), 2001, pp. 349-358.

[27] M.J. Shepperd, C. Schofield, "Estimating software Project Effort using Analogies", in *IEEE TSE*, 23(11), 2000, pp. 736-743.