Journal of Web Engineering, Vol.8, No.2 (2009) 154-181

MEASURES AND TECHNIQUES FOR EFFORT ESTIMATION OF WEB APPLICATIONS: AN EMPIRICAL STUDY BASED ON A SINGLE-COMPANY DATASET

SERGIO DI MARTINO FILOMENA FERRUCCI CARMINE GRAVINO

Dipartimento di Matematica e Informatica, University of Salerno, Italy
{sdimartino | fferrucci | gravino}@unisa.it

EMILIA MENDES

The University of Auckland, Private Bag 92019
emilia@cs.auckland.ac.nz

Received April 12, 2007
Revised August 23, 2008

Effort estimation is a key management activity which goes on throughout a software project being fundamental for accurate project planning and for allocating resources adequately. Thus, it is important to identify techniques and measures that can support such project management activity during the development of Web applications. To this aim, empirical investigations should be performed using data coming from the industrial world. To address this issue, this paper reports on an empirical study based on data from 15 Web applications developed by an Italian software company. The objective of the study was two-fold. The first goal was to verify whether or not some size measures were good indicators of the effort spent to develop the Web applications taken into account. The second goal was to compare the effectiveness of some techniques to establish the relationships between the employed size measures and the development effort of the Web applications. The measures were organized in two sets, where the first one included some length measures while the second one consisted of the nine components which are used to estimate the *Web Objects* measure. The techniques taken into account were *Stepwise Regression*, *Case-Based Reasoning*, and *Regression Tree*. The results indicated that both the sets of size measures were good indicators of the effort for the analyzed dataset. Furthermore, the analysis also revealed that the first set presented significantly superior performance than the second set when using *Stepwise Regression*. No significant differences between the two sets of size measures were highlighted when using *Case-Based Reasoning* and *Regression Tree*.

Key words: Web applications, Size measures, Effort estimation, Empirical validation
Communicated by: D. Lowe & O. Pastor

1 Introduction

Within the even more challenging marketplace of Web applications, the ability to estimate in a reliable way the development effort of these software systems is crucial for the competitiveness of software companies. Indeed, effort estimation is a key management activity which goes on throughout a software project being fundamental for accurate project (re)planning and for allocating resources adequately. In the context of traditional software engineering, several widely accepted (model-based) methods to estimate software development effort can be found. These methods rely on a formal approach, involving the use of algorithms that take in input some project factors influencing the

development effort (such as software size) and produce an effort estimate [12]. However, to date, this is not the case for Web applications, whose development effort is usually estimated using non-model based methods, mostly relying on expert judgments. The main drawback of these approaches lies in the highly-subjectivity, i.e., different experts may provide very different estimations for the Web application to be developed. As a consequence, there is a strong needing for more solid, formal methods to estimate the development effort of Web applications and many researchers are addressing this issue [19, 20, 37-46, 58]. A limitation to the diffusion of model-based methods comes from the lack of widely-accepted ways to estimate Web application size. Indeed, the size measures conceived and widely accepted for traditional software systems, such as *Function Points (FPs)* [5], have been turned out to be inadequate when dealing with Web applications [49, 54, 56, 58]. This has motivated the proposal of several Web application size measures [19, 20, 37, 38, 40, 43, 54]. Nevertheless, to date, there is no standard way to "size" a Web application since relatively few empirical studies have been carried out to validate and compare the different proposals [19, 37, 39, 40, 44, 45, 46]. Analogously, few empirical studies have been carried out in the context of Web applications to verify the effectiveness of the existing techniques to provide an effort estimation starting from some project factors (such as software size) and very few of them have compared the performance of different techniques. This is mainly due to the lack of suitable datasets coming from the industrial world that can be used to perform empirical studies.

The above issues motivated us to collect industrial data to perform an empirical study addressing the following research issues:

> What size measures are good indicators of the effort spent to develop the Web applications of the dataset and what estimation techniques can be considered effective to establish the relationships between the employed size measures and the development effort of the Web applications?

This paper reports on this empirical study, which is based on data from 15 Web applications developed by a single software company. In general, to carry out empirical studies two types of datasets can be taken into account with respect to their source: single-company and cross-company. The former exploits data coming from a single software company while the latter includes information gathered by several software companies. The use of a single-company dataset allowed us to collect data in a controlled and consistent way thus letting us to address one of the most crucial threats to the validity of empirical studies; nevertheless, it prevented us to have a large dataset. This represents the main limitation of the present study that cannot provide definitive results but only offer some indications that should be further validated in subsequent studies.

In our investigation, we employed two sets of size measures: *Set1* included some length measures, such as number of pages, media, server side scripts, etc…that have been indicated by software managers as useful effort predictors, and *Set2* consisted of the nine components used to evaluate the *Web Objects* measure, an extension of *FPs* proposed by Reifer to size Web applications [54]. About the techniques, we applied some of the most used in Software Engineering to estimate development effort, i.e., *Manual Stepwise Regression* [19, 44, 45], *Case-Based Reasoning* [1, 59, 60], a combination of *Regression Tree* and *Stepwise Regression* [10, 11], and a combination of *Regression Tree* and *Case-Based Reasoning* [10, 11].

The remainder of the paper is organized as follows. Section 2 describes the size measures and the dataset used for the empirical study. Section 3 presents the empirical analysis we carried out. Then, a discussion of the empirical results is reported in Section 4, while Section 5 analyses the validity of the empirical study. Related work is provided in Section 6 and some final remarks are given in Section 7.

2 Size Measures and Dataset

In the last years, the Web has become not only a mechanism for sharing information, collaborating and interacting but also a way to access services. The emerging Web technologies leaded to a shift from traditional Web sites, providing navigation mechanisms, to sophisticated and complex Web applications. Nevertheless, the term Web application is sometimes ambiguous and erroneously associated to Web sites and hypermedia applications that are usually smaller software development projects. In the present paper, for Web application we intend a software application characterized by user functionality able to affect the status of the business logic on the Web server, in agreement with [23].

Table 1: Details on the Web applications of the industrial dataset used in the empirical study

Project Number	Project Type	Development Effort (person/hours)	Development Process	Adopted Technologies	Adopted Development Tools
1	e-government (Intranet)	2176	Spiral	J2EE,Oracle	Together J, JBuilder
2	e-government (Intranet/ Internet)	3200	Spiral	J2EE,Oracle	Together J, JBuilder
3	Web Portal	1680	Spiral	ASP .NET, Access	Visual Studio .NET
4	Workflow Management (Intranet)	2024	Spiral	J2EE,Oracle	Together J, JBuilder
5	e-banking	3640	Spiral	J2EE,Oracle	Together J, JDeveloper
6	Web Portal	2792	Spiral	ASP .NET, Oracle	Visual Studio .NET
7	Document management (Intranet)	2768	Rational Unified Process	J2EE,Oracle	Together J, JBuilder
8	e-government	2360	Spiral	J2EE,Oracle	Together J, JBuilder
9	Web Portal	3000	Spiral	ASP .NET, Oracle	Visual Studio .NET
10	Web Portal	1552	Rational Unified Process	J2EE,Oracle	Together J, JBuilder
11	Web Portal	3712	Spiral	ASP .NET, Oracle	Visual Studio .NET
12	Web Portal	3600	Spiral	Oracle, TIBCO, ASP, J2EE	Portal Builder, Together J
13	Web Portal	2800	Spiral	ASP .NET, Oracle	Visual Studio .NET
14	e-government	3688	Spiral	J2EE,Oracle	Together J, JDeveloper
15	Web Portal	1176	Spiral	ASP .NET, Access	Visual Studio .NET

The 15 Web applications employed in our empirical study were e-government, e-banking, Web portals, and Intranet applications. They have been developed by exploiting a wide range of Web-oriented technologies, such as J2EE, ASP.NET, etc… Oracle has been the commonly adopted DBMS, but also SQL Server, Access, and MySQL have been employed in some applications. In all the projects, a Spiral approach was used, except two of them that were managed with the Rational Unified Process. Further details on the projects are provided in Table 1.

Data about the 15 Web applications were provided by an Italian software company, whose core business is the development of enterprise information systems, mainly for local and central government. Among its clients, there are also 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 can be considered a mature software company, having about fifty employees. It is certified ISO 9001, and it is also a certified partner of Microsoft, Oracle, and ESRI.

In our empirical study we took into account two sets of size measures, namely *Set1* and *Set2*. As for the first set, we used both the measures proposed for hypermedia and Web applications by Mendes *et al.*, (such as number of Web pages, new images) [37, 38, 39, 40, 43], and other measures specific of Web applications, such as number of server side scripts/ applications, number of external references, etc… The criteria adopted to select them were: relevance for the designers and developers, easiness to collect, and simplicity and consistency of counting rules [12]. This set is presented in Table 2.

Table 2: The set of measures *Set1*

Variable	**Scale**	**Description**
Web pages (Wpa)	Ratio	Number of static Web pages
Medias (Me)	Ratio	Total number of Multimedia elements in the whole application
New Medias (N_Me)	Ratio	Number of Multimedia elements created from scratch
Client side Scripts and Applications (CSAPP)	Ratio	Number of Client side Scripts and Applications used to provide feature/functionality, dealing with user input (such as a form validator)
Server side Scripts and Applications (SSApp)	Ratio	Number of Server side Scripts and Applications used to modify persistent data and/or to produce a (section of) dynamic Web page basing on some parameters
Internal Links (IL)	Ratio	Number of Internal Links used to connect sections of the Web application
Number of External References (EL)	Ratio	Number of External References used to invoke existing external modules, such as a business tier component or a library routine

As for *Set2*, we took into account the components used to evaluate the *Web Objects* size measure [54]. *Web Objects* extends *FPs* [5] by introducing four Web-related components (*Multi-Media Files*, *Web Building Blocks*, *Scripts*, and *Links*), to be used together with the five traditional *FPs*' function types (*External Input, External Output, External Inquiry, Internal Logical File,* and *External Interface File*) to compute the functional size of a Web application. Reifer devised such list of components

based on the opinion of experts and by analyzing 64 completed Web applications. A description of these predictors is reported in Table 3.

Data about the measures in *Set1* and *Set2* were collected from the analysis and design documents of the 15 Web applications. Table 4 contains the descriptive statistics of the variables composing *Set1* and *Set2* and of the actual development effort, Total Effort (*TotEff*). It is interesting to compare these statistics with the ones of other datasets used in Web engineering researches, such as the Tukutuku [40, 42] or the one used by Ruhe *et al.* [57, 58].

Tukutuku is a cross-company dataset volunteered by software companies all over the World, to develop Web effort estimation models and to benchmark productivity across and within software companies [40, 42]. It consists of data about 68 projects including Web sites and Web applications. For such dataset the mean number of static + dynamic Web pages per project is 37.44 versus 84.13 in our dataset. It is worth noting that also the mean Total Effort of the Tukutuku database (321.33 person/hours per project) is much lower than the mean Total Effort of the dataset used in the present study (2677.87 person/hours per project).

Table 3: The set of measures *Set2*

Variable Name	**Scale**	**Description**
Internal Logical Files (ILF)	Ratio	Number of logical, persistent entities maintained by the Web application to store information of interest
External Interface Files (EIF)	Ratio	Number of logical, persistent entities that are referenced by the Web application, but are maintained by another software application
External Inputs (EI)	Ratio	Number of 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	Number of 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	Number of 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	Number of physical, persistent entities used by the Web application to generate output in multi-media format
Web Building Blocks (WBB)	Ratio	Number of logical persistent entities used to build Web applications and automate their functionality
Scripts (Scr)	Ratio	Number of logical, persistent entities used by the Web application to link internal files and building blocks together in predefined patterns
Links (Lin)	Ratio	Number of logical, persistent entities maintained by the Web application to find links of interest to external applications

Ruhe *et al.* exploited a single-company dataset to verify the effectiveness of *Web Objects* in estimating development effort [47, 48]. The dataset included 12 industrial Web applications with a mean Total Effort of 883 person/hours per project. Thus, the mean effort in our dataset is more than 3 times the one in the Ruhe *et al.*'s dataset (2677.87 vs. 883 person/hours) [47]. Moreover, as for the *Web Objects* components specifically introduced for Web applications by Reifer, let us observe that our dataset presents mean values much greater than the mean values for Ruhe *et al.*'s dataset (in particular, 27.867 *MMF* vs. 15 *MMF*, 100 *WBB* vs. 13 *WBB*, 139.400 *Scr* vs. 6 *Scr*, and 366.800 *Lin* vs. 8 *Lin*).

Table 4: Dataset's descriptive statistics

Variable	Obs	MIN	MAX	MEAN	STD. DEV.
Total Effort (TotEff)	15	1176	3712	2677.867	827.115
Web pages (Wpa)	15	2	46	17.000	12.317
Medias (Me)	15	54	223	104.133	43.500
New Medias (N_Me)	15	20	223	82.533	56.599
Client side Scripts and Applications (CSAPP)	15	5	55	26.933	16.918
Server side Scripts and Applications (SSApp)	15	2	209	80.400	55.414
Internal Links (IL)	15	0	8	4.933	3.770
Number of External References (EL)	15	124	592	279.133	145.322
External Inputs (EI)	15	2	59	24.533	18.302
External Outputs (EO)	15	5	41	20.200	11.965
External Queries (EQ)	15	7	102	40.267	27.044
Internal Logical Files (ILF)	15	0	7	2.733	2.604
External Interface Files (EIF)	15	1	15	5.667	4.624
Multi-Media Files (MMF)	15	12	53	27.867	14.252
Web Building Blocks (WBB)	15	15	225	100.000	49.558
Scripts (Scr)	15	56	260	139.400	62.810
Links (Lin)	15	172	655	366.800	172.825

3 Empirical Study

The empirical study was aimed to assess both the size measures described in the previous section and some effort estimation techniques. In particular, we employed *Stepwise Regression* (*SWR*), *Case-Based Reasoning* (*CBR*), and *Regression Tree* (*RT*), which have been extensively adopted in the context of effort estimation for traditional software and have been recently applied also in the context of Web effort estimation [6, 19, 20, 37-46, 57-60].

In the following, we first provide an overview of the effort estimation techniques we used and of the criteria we adopted to assess their accuracy. Then, we describe the application of the techniques. A discussion of the results is provided in Section 4.

3.1 Employed effort estimation techniques

Stepwise Regression (*SWR*) is a statistical technique whereby a prediction model (an equation) is built that represents the relationship between independent (e.g. number of Web pages) and dependent variables (e.g. Total Effort) [35]. This technique allows us to compute linear regression in stages [48]. Indeed, the model is built by adding, at each stage, the independent variable with the highest association to the dependent variable, taking into account all the 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). Different approaches have been proposed so far to apply *SWR*. In the current study, we applied a *Manual forward SWR* (*MSWR*), using the technique proposed by Kitchenham [28]. Basically, we used this technique to select the most influencing independent variables and then we performed the linear regression to obtain the final model. To evaluate the goodness of fit of the obtained regression model we used the *adjusted* R^2, the square of the linear

correlation coefficient, indicating the amount of the variance of the dependent variable explained by the model related to the independent variables. The *adjusted* R^2 is a modification of R^2 that adjusts for the number of explanatory terms in a model. Unlike R^2, the *adjusted* R^2 increases only if the new term improves the model more than would be expected by chance. Other useful indicators taken into account were the *p-values* and *t-values* for the coefficients and the intercept of the obtained model. The *p-values* give an insight into the accuracy of the coefficients and the intercept, whereas their *t-values* allow us to evaluate their importance for the generated model. In particular, *p-values* less than 0.05 are considered an acceptable threshold, meaning that the variables are significant predictors with a confidence of 5%. As for the *t-value*, a variable is significant if its corresponding value is greater than 1.5.

Case-Based Reasoning (*CBR*) is an Artificial Intelligence technique that allows us to determine the effort of a new project (*target case*) by considering some similar projects previously developed (*case base*) [59]. In particular, for our empirical study, once the projects have been characterized in terms of measures in *Set1* (resp. *Set2*) the similarity between the target case and the other cases is measured, and the most similar cases are used, possibly with adaptations, to obtain the estimation.

Regression Tree (*RT*) is a variant of decision trees that can be used to approximate real-valued functions [10, 11, 39]. This technique takes as input a set of numerical variables and generates a binary tree estimating the value of the target variable (corresponding to the dependent variable in linear regression). In particular, the leaves of the binary tree suggest the values for the target variable on the base of the values of the predicting variables (the independent variables in linear regression). The binary tree is built by recursively splitting the input data (i.e., the values of the predicting variables) into partitions. At the beginning, all data are associated to the root. Then, they are split in two parts, minimizing the sum of the squared deviations from the mean in the separated parts. At each split, the process determines the input variable to be used for splitting, and its values to associate to the left and right child nodes, respectively. Let us observe that each node has associated the mean value of the target variable. The process ends when, for each node, a minimum size, specified for the node by the user, is obtained. Subsequently, to determine the predicted value for the target variable, we start from the root node and then follow the right or left branch, based on the value of the splitting variable. We continue until a leaf node is reached, which contains the predicted value.

3.2 Evaluation criteria

To assess each technique with *Set1* and *Set2*, we applied a *leave-one-out cross-validation*, which is widely used in the literature when dealing with small datasets (see, e.g. [10, 23]). To apply the cross-validation, the original dataset is divided into *n* different subsets (where *n* is the size of the original dataset) of *training* and *validation* sets, where each validation set has one project. Then, *n* steps are performed to get the predictions for the *n* validation sets. At each step, for *MSWR*, the training set is employed to determine the prediction model that is used to determine the effort prediction for the validation set. The equivalent for *CBR* is to use the training set as a case base, and then to estimate effort for the project that has been removed. As for *RT*, at each step of the cross-validation the training set is used to build the binary tree while the validation set is used to obtain the prediction.

At each step of the leave-one-out cross-validation we evaluated the accuracy of each prediction by calculating the corresponding *absolute residual* and *Magnitude of Relative Error* (*MRE*). These are

two classical measures: the absolute residual is the absolute value of the difference between actual effort and estimated effort, while *MRE* is defined as:

$$MRE = \frac{\left|e - \hat{e}\right|}{e} \qquad (1)$$

where e represents actual effort and $\hat{e}$ estimated effort.

Thus, these measures provide an insight on the precision of a single estimation. To assess the precision of all the estimations derived during the leave-one-out cross validation, we used some summary measures, namely the *Mean of MRE* (*MMRE*), *Median of MRE* (*MdMRE*), and *Pred*(*0.25*) [12], together with *boxplots of absolute residuals* [32]. In the following we briefly recall their main underlying concepts.

To have a cumulative measure of the error, the *MRE* values have to be aggregated across all the observations of the leave-one-out cross-validation. We used the mean and the median, two measures of the central tendency, giving rise to *MMRE* and *MdMRE*, where the latter is less sensitive to extreme values [39]. According to Conte *et al.* [17], a good effort prediction model should have a $MMRE \leq 0.25$, to denote that the mean estimation error should be less than 25%.

Pred(*n*) measures the percentage of estimates that are within *n*% of the actual values. In other words, *Pred*(0.25) is the percentage of predictions whose error is less than 25%. Again, according to Conte *et al.*[17], a good prediction approach should present a $Pred(0.25) \geq 0.75$, meaning that at least 75% of the predicted values should fall within 25% of their actual values.

Boxplots of absolute residuals are widely employed in exploratory data analysis since they provide a quick visual representation to summarize data using five numbers: median, upper and lower quartiles, minimum and maximum values, and outliers. The box of the plot is a rectangle with an end at each quartile and a line is drawn across the box at the sample median (*m* in Figure 1). The lower quartile (*l* in Figure 1) is determined considering the bottom half of the data, below the median, i.e., by finding the median of this bottom data, while, the upper quartile (*u* in Figure 1) is the median of the upper half of the data, above the median. The length of the box *d* is the inter-quartile range of the statistical sample. Lower tail is $u+1.5*d$ while $u-1.5*d$ is the Upper tail. Points at a distance from the median greater than 1.5 times the inter-quartile range represent potential outliers and are plotted individually.

Moreover, we tested the *statistical significance* of the obtained results by using absolute residuals, in order to establish if one of the set of measures, and of the employed estimation techniques, provided better results than the others [25, 31]. In particular, we performed statistical test to verify the following Null Hypothesis "the two considered population have identical distributions". This kind of test is used to verify the hypothesis that the mean of the differences in the pairs is zero. The test statistic is the number of positive differences. If the null hypothesis is true, then the number of positive and negative differences should be approximately the same.

In order to have also an indication of the practical/managerial significance of the results we verified the *effect size* [27]. Effect size is a simple way of quantifying the difference between two groups. It 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"

[27]. Effect size emphasises the size of the difference rather than confounding this with sample size. It is just the standardized mean difference between the two groups (i.e., mean difference/standard deviation) and it is exactly equivalent to a 'Z-score' of a standard Normal distribution. For this reason, employing the Wilcoxon test, the effect sizes have been 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 [15]: so r=0.20 indicates a small effect, r=0.50 indicates medium effect, and r=0.80 indicates a large effect.

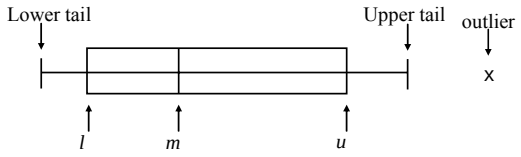

Figure 1: The Boxplot

Finally, as suggested by Mendes and Kitchenham [44, 45], we also analyzed the values of *MMRE*, *MdMRE*, and *Pred*(0.25) obtained by employing at each step of the leave-one-out cross-validation the mean of effort and the median of effort as the estimation. The aim of this analysis is to assess whether the estimations obtained with *MSWR*, *CBR*, and *RT* are significantly better than the estimations based on the mean or median effort. If this assumption is not verified, then a software company could simply base its estimations on the mean or the median of previous project efforts, in place of the more advanced (and time-consuming) techniques, such as *MSWR*, *CBR*, and *RT*.

In the following, we show the application of all the described techniques and evaluation criteria on our dataset.

3.3 Applying forward stepwise regression

In order to apply *MSWR* we had to select the important independent variables to be used in the prediction model. It is worth noting that we did not perform a separate selection of variables for each leave-one-out cross-validation step; but for sake of simplicity we rather performed a regression using the variables previously selected applying *MSWR*. Thus, we first employed the whole dataset of 15 Web applications (1) to verify the assumptions underlying the *Stepwise Regression*, (2) to check the presence of possible outliers, (3) to select the variables, and (4) to analyze the fitting of the obtained model and whether or not the selected variables could be considered good indicators of the effort. The details on the analysis carried out to verify these points and the models obtained are reported in the Appendix. In particular, the application of *MSWR* with *Set1* identified three attributes as the main factors affecting the development effort: the number of Server-side Scripts and Applications (*SSApp*), the number of Internal Links to other components (*IL*), and the number of Multimedia elements (*Me*). As for *Set2* the External Inputs (*EI*) variable was identified as the main factor affecting the development effort. Using these variables, we performed a leave-one-out cross-validation to assess the accuracy of the estimations obtained with the *MSWR* technique using the *MMRE*, *MdMRE*, and *Pred*(0.25) summary measures which are reported in Table 5. These statistics are good if we assume as reasonable thresholds Conte *et al.*'s suggestion [17] (except *MSWR* with *Set2* that presents a *Pred*(0.25) slightly less than 75%). The table also reports the results presented in [19] where we

employed the same dataset but a different application of *SWR*, namely the *SWR* procedure implemented in the SPSS[a] tool and not the manual one used in the current study.

As we can observe, *MSWR* produced much better results than *SWR* with *Set1*, while for *Set2* we obtained the opposite situation. Moreover, observe that *SWR* applied in [19] selected different variables, namely *Wpa* and *Me* from *Set1*, and *EI* and *Lin* from *Set2*.

Table 5: Results for *Manual Stepwise Regression*

Technique	**Measures**	***MMRE***	***MdMRE***	***Pred(0.25)***
MSWR	Set1	12%	11%	87%
	Set2	23%	21%	73%
SWR (SPSS) [19]	Set1	28%	16%	53%
	Set2	16%	9%	80%

3.4 Applying Case-Based Reasoning

CBR determines the effort of a new application by comparing it with similar projects previously developed, [26, 59]. To use the method, some choices have to be done: the appropriate similarity function, the number of analogies to select the similar projects to consider, the analogy adaptation strategy for generating the estimation, and the relevant project features. We used the ANGEL tool [59] that about the similarity function implements the Euclidean distance using variables normalized between 0 and 1. Selecting the number of analogies is a key task, since it refers to the number of similar cases to use for estimating the effort required by the target case. Since we dealt with a small dataset, we used 1, 2, and 3 analogies, as suggested in other similar works [10, 39]. Moreover, to select similar projects for the estimation, we employed as adaptation strategies the *mean of k analogies* (*simple average*), the *inverse distance weighted mean*, and the *inverse rank weighted mean* [59]. As for the selection of the features, we used *Feature Subset Selection* (*FSS*) of ANGEL in order to let the tool to automatically choose, among all the variables of Table 2 and Table 3, the ones to employ as set of key features in the analogy-based estimation. This technique looks for the optimal feature subset with an exhaustive search [26]. As alternative technique, we also used the *Pearson's Correlation* to select variables to be considered in the application of *CBR*. This was carried out to verify if this approach gives better results than the ANGEL's *FSS* automatic feature. In particular, we applied it to identify all the variables that were statistically correlated to the effort at level 0.05. Table 6 shows the selected variables for each combination of the employed number of analogies and adaptation strategy, using *FSS* of ANGEL, while Table 7 reports the variables we obtained with the *Pearson's Correlation*. To discern among all these combinations, we used the following naming conventions:

- CBR_i denotes *CBR* with *i* analogies.
- A subsequent *A* (CBR_{iA}) denotes the use of mean of *k* analogies as adaptation strategy, a *B* (CBR_{iB}) the use of inverse distance weighted mean, and a *C* (CBR_{iC}) the inverse rank weighted mean.

[a] SPSS vers.13.0 has been used to carry out the statistical analysis performed with SWR.

- A further "*-FSS*" ($CBR_{1A\text{-}FSS}$) indicates that we used the ANGEL's *FSS* characteristic, while a "*-PC*" ($CBR_{1A\text{-}PC}$) denoted the use of *Pearson's Correlation* to select the variables.

ANGEL calculated 15 predictions and the corresponding residuals for each selection of the number of analogies and of the analogy adaptation techniques. Each prediction was obtained by selecting a target project from the dataset and by considering as case base the other 14 projects.

Table 6: The variables selected by using the *Feature Subset Selection* of ANGEL

	Set1	**Set2**
$CBR_{1A\text{-}FSS}$	SSA, EL, Me, IL	EI, ILF, SCR
$CBR_{2A\text{-}FSS}$	SSA, Me, N_Me, CSApp	EI, EQ, ILF, EIF, SCR
$CBR_{3A\text{-}FSS}$	SSA, Me, N_Me, CSApp	EI, EQ, EIF, SCR
$CBR_{2B\text{-}FSS}$	SSA, EL, Wpa, Me, N_Me, CSApp	EI, EQ, EIF, SCR
$CBR_{3B\text{-}FSS}$	SSA, EL, N_Me, CSApp	EI, EIF, SCR
$CBR_{2C\text{-}FSS}$	SSA, EL, Me, CSApp, IL	EI, ILF, SCR
$CBR_{3C\text{-}FSS}$	SSA, Me, N_Me, CSApp	EI, EQ, EIF, SCR

Table 7: The variables selected by using the *Pearson's Correlation* test

	Variables	**p-value**
Set1	SSA	0.008
	EL	0.034
Set2	EI	0.001
	EQ	0.014
	SCR	0.012
	Lin	0.002

Table 8: Results for *Case-Based Reasoning*

Technique	**Measures**	**CBR Configuration**	***MMRE***	***MdMRE***	***Pred(0.25)***
CBR (with variable selection)	Set1	$CBR_{2C\text{-}FSS}$	19%	9%	87%
		$CBR_{2C\text{-}PC}$	25%	17%	67%
	Set2	$CBR_{2B\text{-}FSS}$	11%	10%	93%
		$CBR_{2A\text{-}PC}$	24%	17%	87%
CBR (no variable selection) [19]	Set1	CBR_{2B}	22%	13%	73%
	Set2	CBR_{3B}	24%	12%	73%

Table 8 reports the best results, in terms of *MMRE*, *MdMRE*, and *Pred*(0.25), obtained with *FSS* of ANGEL and *Pearson's Correlation* test to select variables. Note that Table 8 also presents the results obtained in [19], where *CBR* was applied without using the selection of variables. We can observe that for both *Set1* and *Set2* the best results were obtained by using ANGEL's *FSS* to select the relevant project features and employing 2 analogies. In particular, for *Set1* the best result was obtained when

the inverse rank weighted mean was used as adaptation strategy, while for *Set2* with the inverse distance weighted mean.

As we can note, also *CBR* can be considered effective assuming as reasonable thresholds Conte *et al.*'s suggestions, except for $CBR_{2C\text{-}PC}$, CBR_{2B}, and CBR_{3B} that present *Pred*(0.25)= 67%, *Pred*(0.25)= 73%, and *Pred*(0.25)= 73%, respectively. It is worth noting that the results obtained with ANGEL's *FSS* were better than those obtained by selecting the variables with the *Pearson's Correlation*. Moreover, the automatic selection of the variables led to improved predictions if compared to those presented in [19], where no selection was used.

3.5 Applying Regression Tree

In literature, some authors suggest to use *RT* only in presence of large datasets [10, 11]. If this is not the case, *RT* should be used in combination with other techniques, to partition the dataset into more heterogeneous groups. We followed this approach, by applying *RT* together with *CBR* and *MSWR*, as detailed in the following.

The application of *RT* on *Set1* determined the splitting of the group on the variable *SSApp* (see Figure 2(a)). Thus, this method suggested the number of server-side scripts and applications to be employed as predictor for obtaining the estimations. Indeed, the left child node 2 (mean *TotEff*=1962.3) can be reached if *SSApp* ≤ 63 and has 7 rows associated, while the right child node 3 (mean *TotEff* = 3304) can be reached if *SSApp*>63 and has 8 rows associated. On the other hand, when considering *Set2*, the technique split the group on the variable *EI* (see Figure 2(b)). Again, as for *MSWR* also this method suggested to use the number of external inputs for obtaining the estimations. In this case the left child node 2 (mean *TotEff* = 2024) has 7 rows associated and a case goes left if *EI* ≤ 17. The right child node 3 (mean *TotEff* = 3250) has 8 rows and a case goes right if *EI* > 17.

Since both of them were trivial splits for the effort prediction, they were used together with *CBR* (and *MSWR*), as suggested by Briand *et al.* in [11]. In particular, regarding the first combination, we applied *CBR* using *FSS* on the two sets of observations associated to the leaf nodes of the trees obtained with *RT* (shown in Figure 2).

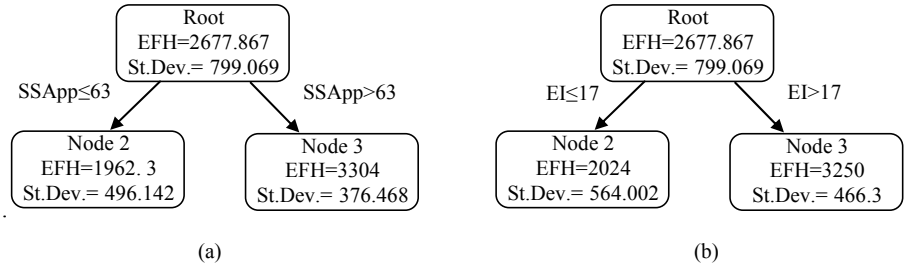

Figure 2: The regression trees for *Set1* (a), and *Set2* (b)

Table 9 shows the variables selected for each combination of number of analogies and adaptation strategy employed. Note that we used the same naming convention as in the previous subsection, adding the prefix "*RT*+" to denote the application of *Regression Tree*.

Table 9: The variables selected by using ANGEL's *Feature Subset Selection* on the subsets of cases obtained with *Regression Tree*

Technique	**Set1**	**Set2**		
	Cases with SSApp ≤ 63	**Cases with SSApp > 63**	**Cases with EI ≤ 17**	**Cases with EI > 17**
RT+CBR$_{1A\text{-}FSS}$	*EL, CSApp, IL*	*Me*	*EI, ILF, MMF, Lin*	*EIF, Scr*
RT+CBR$_{2A\text{-}FSS}$	*Me, CSApp*	*Me*	*EI, Scr*	*EQ, EO, EIF*
RT+CBR$_{3A\text{-}FSS}$	*CSApp, IL*	*EL, Me, CSApp, IL*	*EI, EIF, Scr*	*EI, EQ, EO*
RT+CBR$_{2B\text{-}FSS}$	*EL, N_Me, CSApp, IL*	*Me*	*EI, Scr*	*EIF, MMF, Scr*
RT+CBR$_{3B\text{-}FSS}$	*SSA, N_Me, CSApp, IL*	*Me*	*EI, EIF, Scr*	*EIF, Scr*
RT+CBR$_{2C\text{-}FSS}$	*EL, CSApp, IL*	*Me*	*EI, Scr*	*EIF, Scr*
RT+CBR$_{3C\text{-}FSS}$	*SSA, N_Me, CSApp, IL*	*Me*	*EI, EIF, Scr*	*EI, EQ, EO, MMF*

The statistics for the combination of *RT* and *CBR* techniques are reported in Table 10, highlighting very positive results. Table 12 also includes the results we obtained in [19], where *RT* was combined with *CBR* without considering the selection of variables. We can observe that the combination performed in the current study confirmed that the use of *FSS* when applying *CBR* provides better results than when *FSS* is not used.

Table 10: Results for *Regression Tree* + *Case-Based Reasoning*

Technique	**Measures**	**CBR Configuration**	***MMRE***	***MdMRE***	***Pred(0.25)***
RT + *CBR* (with variable selection)	Set1	RT + CBR$_{2C\text{-}FSS}$	11%	09%	93%
	Set2	RT + CBR$_{1A\text{-}FSS}$	19%	20%	66%
RT + *CBR* (without variable selection) [19]	Set1	RT + CBR$_{2B}$	17%	13%	87%
	Set2	RT + CBR$_{2B}$	26%	19%	72%

Conversely, the application of *Regression Tree* in combination with *Linear Regression* did not provide good models, again confirming the results of [19]. In particular, all the models were characterized by very low *adjusted* R^2 and the variables (i.e., *SSApp* and *EI*) did not present good *t-values* and *p-values*. Due to these negative results, their models are omitted.

4 Discussion and comparison of results

Let us recall that the research question that we were addressing in our empirical study was:

> What size measures are good indicators of the effort spent to develop the Web applications of the dataset and what estimation techniques can be considered effective to establish the relationships between the employed size measures and the development effort of the Web applications?

The results in terms of *MMRE*, *MdMRE*, and *Pred*(0.25), reported in Table 5, 8, and 10, indicated that both sets of measures *Set1* and *Set2* were good indicators of the development effort with each one of the employed estimation techniques, since they suit (or were very close to) Conte *et al.*'s thresholds

[17]. In particular, *MSWR* highlighted that the number of Server-side Scripts and Applications (*SSApp*), the number of Internal Links to other components (*IL*), and the number of Multimedia elements (*Me*) were the most influencing factors among *Set1*, while External Input (*EI*) was the most influencing factor among *Set2*. Moreover, the analysis of *MMRE*, *MdMRE*, and *Pred*(0.25) revealed that the best results with *Set1* came from the application of the combination of *RT* and *CBR*, using *FSS* with 2 analogies and inverse rank weighted mean as adaptation strategy. As for *Set2*, the best results were related to *CBR* using *FSS*, with 2 analogies and inverse distance weighted mean as adaptation strategy.

The boxplots of absolute residuals presented in Figure 3 graphically confirmed the results obtained with *MMRE*, *MdMRE*, and *Pred*(0.25). In particular, they showed that the spread of the distributions for *CBR* using *Set1* were slightly wider than those for *CBR* using *Set2*. Conversely, the spread of the distribution for *MSWR* using *Set2* were slightly wider than those for *MSWR* using *Set1*; however *MSWR* boxplot presented two outliers. By analyzing the medians, we can note that *CBR* with *Pearson's Correlation* test on *Set1* presented the largest residuals, followed by *MSWR* and *CBR* with *Pearson's Correlation* test on *Set2*.

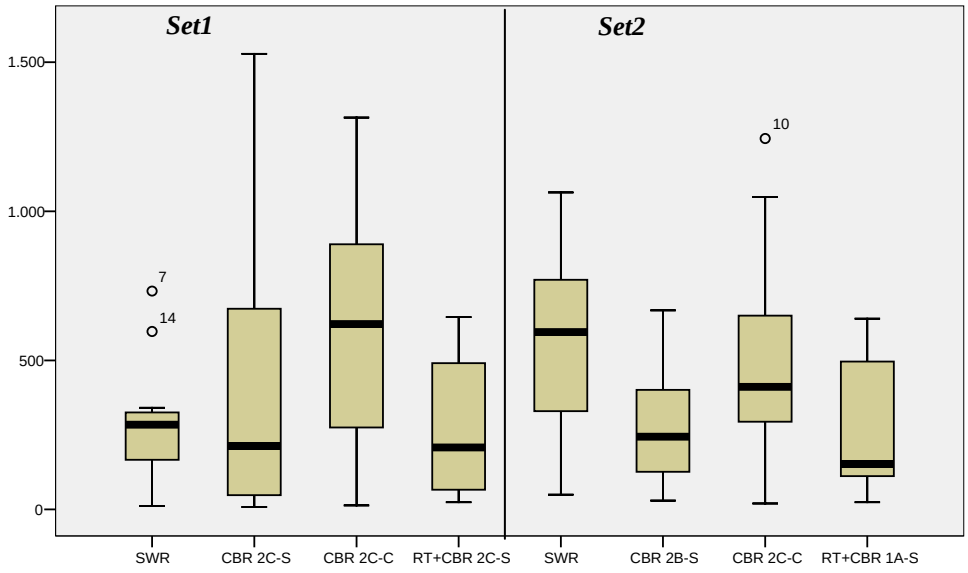

Figure 3: Boxplots of absolute residuals

In order to establish if one of the set of measures and of the employed estimation techniques provided significantly better results than the others, we checked whether or not there were statistically significant differences between the obtained absolute residuals [32, 39] applying a non-parametric test – the Wilcoxon test[b]. The analysis of the results (see Table 11) revealed that *MSWR* with *Set1* presented significantly superior performance than *Set2*. Observe that a different situation was obtained

[b] We used a non parametric test since the residuals were not normally distributed in three cases ($CBR_{2C\text{-}FSS}$ and $RT{+}CBR_{2C\text{-}FSS}$ with *Set1*, $RT{+}CBR_{1A\text{-}FSS}$ with *Set2*).

with *SWR* in [19], where *Set2* provided better results than *Set1*. On the other side, this analysis did not report any significant difference between the two sets of size measures, when using *CBR* with *FSS*, *CBR* with *Pearson's Correlation* test, and *RT+CBR*. When comparing residuals of *MSWR*, *CBR*, and *RT+CBR*, Wilcoxon test revealed that using *Set2*, *CBR* with *FSS* provided significantly better results than *MSWR* and *CBR* with *Pearson's Correlation* to select variables, *RT+CBR* with *FSS* provided significantly better results than *MSWR* and *CBR* with *Pearson's Correlation* to select variables. On the other hand, using *Set1*, *MSWR* provided significantly better results than *CBR* with *Pearson's Correlation* to select variables, *RT+CBR* with *FSS* provided significantly better results than *CBR* with *Pearson's Correlation* to select variables. No other significant differences among the different techniques were revealed for the two set of variables.

Table 11. Summary of the Wilcoxon test and effect sizes

		Set1				**Set2**			
		MSWR	*CBR-FSS*	*RT+ CBR-FSS*	*CBR-PC*	*MSWR*	*CBR-FSS*	*RT+ CBR-FSS*	*CBR-PC*
Set1	*MSWR*		=	=	> (0.55)	> (0.77)	=	=	> (0.56)
	CBR-FSS			=	=	=	=	=	=
	RT+ CBR-FSS				> (0.56)	> (0.56)	=	=	> (0.57)
	CBR-PC					=	< (0.57)	< (0.61)	=
Set2	*MSWR*						< (0.74)	< (0.75)	=
	CBR-FSS							=	> (0.55)
	RT+ CBR-FSS								> (0.60)
	CBR-PC								
MTotEff		< (0.75)	=	< (0.74)	=	=	< (0.70)	< (0.73)	=
MdTotEff		< (0.58)	=	< (0.56)	=	=	< (0.53)	< (0.57)	=

- ">" means that "the technique indicated on the row provided significantly better estimations than the one on the column"
- "<" means that "the technique indicated on the row provided significantly worst estimations than the one on the column"
- "=" means that there is no significant difference between the techniques on the row and column
- The practical significance in terms of the effect size is reported between brackets.

Moreover, to have an indication of the practical/managerial significance of the results we also analyzed the effect size [27]. In our analysis, the statistics on effect size revealed that all results statistically significant were also practical significant according to the widely used Cohen's benchmarks [15]. Indeed, medium effect sizes were highlighted for:

MSWR with *Set1* vs CBR_{PC} with *Set1* (r=0.55);

MSWR with *Set1* vs *MSWR* with *Set2* (r=0.77);

MSWR with *Set1* vs CBR_{PC} with *Set2* (r=0.56);

$RT+CBR_{FSS}$ with *Set1* vs CBR_{PC} with *Set1* (r=0.56);

$RT+CBR_{FSS}$ with *Set1* vs *MSWR* with *Set2* (r=0.87);

$RT+CBR_{FSS}$ with *Set1* vs CBR_{PC} with *Set2* (r=0.57);

CBR_{PC} with *Set1* vs CBR_{FSS} with *Set2* (r=0.57);

CBR_{PC} with *Set1* vs $RT+CBR_{FSS}$ with *Set2* (r=0.61);

MSWR with *Set2* vs CBR_{FSS} with *Set2* (r=0.74);

MSWR with *Set2* vs $RT+CBR_{FSS}$ with *Set2* (r=0.75);

CBR_{FSS} with *Set2* vs CBR_{PC} with *Set2* (r=0.55);

$RT+CBR_{FSS}$ with *Set2* vs CBR_{PC} with *Set2* (r=0.60).

We also compared the results with the predictions obtained with *MTotEff* and *MdTotEff* techniques, which use as estimation simply the mean of effort and the median of effort of all the developed Web applications, respectively. Table 12 reports on the summary statistics *MMRE*, *MdMRE*, and *Pred*(0.25) for the application of the leave-one-out cross validation with these techniques. As we can easily observe these statistics are quite far from the thresholds suggested by Conte *et al.* [17]. Moreover, the Wilcoxon test on the absolute residuals also showed that the results obtained using *Set1* with *MSWR* (*RT+CBR* with *FSS*, resp.) were significantly better than those obtained using *MTotEff* and *MdTotEff*, whereas the results obtained using *Set2* and *CBR* with *FSS* (*RT+CBR* with *FSS*, resp.) were significantly better than those obtained using *MTotEff* and *MdTotEff*. These results were also confirmed by the effect size analysis. Indeed, medium effect sizes were highlighted for:

MTotEff vs *MSWR* with *Set1* (r=0.75);

MdTotEff vs *MSWR* with *Set1* (r=0.58);

MTotEff vs $RT+CBR_{FSS}$ with *Set1* (r=0.74);

MdTotEff vs $RT+CBR_{FSS}$ with *Set1* (r=0.56);

MTotEff vs CBR_{FSS} with *Set2* (r=0.70)

MdTotEff vs CBR_{FSS} with *Set2* (r=0.53)

MTotEff vs $RT+CBR_{FSS}$ with *Set2* (r=0.73)

MdTotEff vs $RT+CBR_{FSS}$ with *Set2* (r=0.57)

Table 12: Results for *MTotEff* and *MdTotEff*

Technique	***MMRE***	***MdMRE***	***Pred(0.25)***
MTotEff	0.34	0.27	0.47
MdTotEff	0.33	0.24	0.60

5 Validity

It is widely recognized that several factors can bias the *construct*, *internal*, *conclusion*, and *external validity* of empirical studies. To satisfy *construct validity* a study has "to establish correct operational measures for the concepts being studied" [29]. In other words, it represents to what extent the predictor and response variables precisely measure the concepts they claim to measure [39]. A research study has *internal validity* if its outcome can be considered as a function of the measured variables and not of extraneous factors. *Conclusion validity* is concerned with the ability to draw statistically correct conclusions while *external validity* is concerned with the ability to generalize the results to other contexts. In the following, we will analyze the presented empirical study with respect to these types of validity so that the reader is aware of its strengths and weakness.

5.1 Construct validity

As for the construct validity, the choice of the size measures and how to collect them represents the crucial aspects.

Regarding the selection of the size measures, the criteria we adopted to define *Set1* were: relevance for the designers and developers, easiness to collect, and simplicity and consistency of counting rules [12]. They were indicated by software managers as useful indicators to provide cost estimations and most of them have been used in other empirical studies [37, 38, 39, 40, 43]. *Set2* are the components to determine *Web Objects,* a size measure previously used in the literature by other researchers (e.g., [19, 54, 57, 58]).

In order to verify the validity and reliability of the employed measures to size Web applications we performed a factor analysis [24], a technique quite popular in social sciences to assess construct validity (see e.g. [47]). It has also been used in empirical software engineering (see e.g. [21, 22, 51]). This type of analysis allows one to derive not-so-observable factors from a collection of observable variables. Thus, it has been investigated herein to highlight the underlying factors of Web application size that are measured by the employed *Set1* and *Set2* measures and how these measures load on the identified factors. To carry out the analysis we used the SPSS tool with *principal component extraction* and *varimax rotation* [24] [47]. In order to apply the technique properly and to interpret the results we verified several indices, such as *KMO (Kaiser_Meyer_Olkin)-measure*, *mean of communalities*, and *Bartlett's test of sphericity*, which mainly assess the degree of correlation among the considered variables [47]. The results of these tests revealed that a factor analysis was indeed possible. For that analysis we had to select the thresholds for factor loadings and extracted variance. A factor loading represents the correlation between a variable (each measure in *Set1* and *Set2*) and a factor. Comrey suggested that loadings in excess of 0.45 could be considered fair, those greater than 0.55 as good, those of 0.63 very good, and those of 0.71 as excellent [14]. Thus, we took into account factor loadings

greater than 0.63. As for extracted variance, in order to select the variables that contribute substantially to the variance of the factors, we considered those that exceed 0.50 [33].

Tables 13 and 14 show respectively the resulting matrix of factor loadings, after varimax rotation for the sets of variables *Set1* and *Set2*.

Concerning *Set1*, three factors have been identified (see Table 13) with approximately 88% of the variance explained and factor loadings greater than 0.63. In particular, variables *Me*, *N_Me*, and *EL* load on Factor 1, variables *Wpa*, *CSApp*, and *IL* load on Factor 2, and *SSApp* loads on Factor 3. A quite intuitive interpretation can be provided for Factors 1, 2 and 3. Factor 1 seems to characterize multimedia elements (excluding external references); Factor 2 seems to characterize client side components of a Web application (excluding multimedia elements), while Factor 3 seems to represent server side components.

Table 13. The matrix of factor loadings after varimax rotation for the variables in *Set1*

	Factor 1	**Factor 2**	**Factor 3**
Web pages (Wpa)		**0.797**	
Media (Me)	**0.935**		
New Media (N_Me)	**0.892**		
Client side Scripts and Applications (CSAPP)		**0.943**	
Server side Scripts and Applications (SSApp)			**0.958**
Internal Links (IL)		**0.831**	
Number of External References (EL)	**0.853**		
Total Eigenvalue	2.449	2.416	1.302
% of Variance	0.350	0.345	0.186
Cumulative Variance	0.350	0.695	0.881

Table 14. The matrix of factor loadings after varimax rotation for the variables in *Set2*

	Factor 1	**Factor 2**
External Inputs (EI)		**0.651**
External Outputs (EO)		**0.752**
External Queries (EQ)		**0.890**
Internal Logical Files (ILF)	**0.928**	
External Interface Files (EIF)	**-0.930**	
Multi-Media Files (MMF)		**0.685**
Web Building Blocks (WBB)	**0.847**	
Scripts (Scr)	**0.856**	
Links (Lin)		**0.977**
Total Eigenvalue	3.634	3.390
% of Variance	0.404	0.377
Cumulative Variance	0.404	0.780

Regarding *Set2*, two factors have been identified (see Table 14) with approximately 78% of the variance explained and factor loadings greater than 0.63. The interpretation of these factors is quite intuitive. Indeed, the variables loading on Factor 1, namely *EIF*, *ILF*, *Scr*, and *WBB*, are related to the business logic of a Web application. The second factor is composed by the variables *Lin*, *EQ*, *EO*, *MMF* and *EI*, which are all involved in the presentation layer of a Web application.

A positive aspect resulting from the analysis is that for both *Set1* and *Set2*, each variable loads on one and only one factor, which is crucial for construct validity [21, 22].

Now, we discuss the other crucial aspect related to construct validity, namely the collection of information about the measures and the actual effort [36]. Concerning the effort collection, the software company kept track of the effort spent for each project through a controlled process, where each team member daily annotated the information about his/her development effort and weekly each project manager stored the sum of the efforts for the team.

In order to collect all the significant information to calculate the values of the size measures, the authors defined a template to be filled in by the project managers. All the project managers were trained on the use of the questionnaires. Moreover, they employed the counting conventions of *FPs* [25] and followed the suggestions provided by Reifer in his "Web Objects White Paper" [47] where he explains how to quantify the *Web Objects* components for a given Web application. The projects managers had experience in measuring functional measures such as *FPs* so they did not manifest any kind of difficulties to collect the information required for our empirical analysis. One of the authors analyzed the filled templates and the analysis and design documents, in order to cross-check the provided information.

Thus, we made all the possible to perform the data collection in a controlled and uniform fashion, in order to ensure accuracy of the results.

5.2 Internal validity

Some factors should be taken into account for the internal validity: subjects' authoring and designing experience, reliability of the data and lack of standardization [6, 7, 28, 29, 37, 39].

The subjects involved in the study were professionals who worked in the software company. 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.

As for the reliability of the data and lack of standardization, the adopted questionnaires were the same for all the projects and the project managers were instructed on how to use the questionnaires to correctly provide the required information. Instrumentation effects in general did not occur.

5.3 Conclusion validity

As for the conclusion validity we carefully applied the statistical tests, verifying all the required assumptions. However, the main threat to conclusion validity is related to the number of projects composing the dataset. Indeed, 15 projects can be considered a quite small number from a statistical point of view to allow us to make generalization about the values obtained (in particular with *Stepwise*

Regression). However it is recognized that getting information on a great number of real world software applications is quite hard [11, 34, 50, 59] and this is especially true for Web applications for several reasons [42]. For example, for a dataset obtained from a single company problems that can occur are [11, 42]:

i. the time required to accumulate enough data on past projects from a single company may be prohibitive;

ii. by the time the dataset is large to be of use, technologies used by the company may have changed, and older projects may no longer be representative of current practices.

Though the use of cross-company datasets helps researchers to overcome the difficult to collect a significant large dataset, it can introduce different kinds of biases and risks. In particular:

i. It is more difficult to ensure that data are collected in a consistent manner by means of a uniform data collection control across different companies.

ii. It is more difficult to ensure that project data represent a sample representative of a well-defined population (with well-defined characteristics).

iii. Differences in processes and practices may result in trends that may differ significantly across companies.

iv. Furthermore, it is recognized in the literature that models built using cross-company datasets provide much less accurate estimations than the ones obtained using single-company datasets [42, 44, 45].

For these reasons, several case studies are reported in the literature based on few observations (see e.g., [50] exploiting 9 observations and [34, 63] exploiting 19 observations). Indeed, this kind of study, though cannot provide general results, contributes to offer useful indications that can be further validated in subsequent studies. So, the reader should be aware that other investigations should be performed to verify/confirm the empirical results presented in this study.

5.4 External validity

The applications involved in this empirical analysis are representative samples of modern Web applications, taking into account their type, functionalities, target platforms, and complexity. Thus, we are confident that the type of analyzed Web applications did not bias the validity of the achieved results.

On the other hand, 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 [13]. Thus, replications of the study taking into account data from other companies are necessary to get a generalization of the results.

6 Related Work

Several empirical studies have been carried out so far to analyze the effectiveness of *Stepwise Regression*, *Case-Based Reasoning*, and *Regression Tree* in the case of traditional software

applications. In particular, Briand *et al.* applied *Linear* (*Stepwise*) *Regression*, *Regression Tree*, and *Case-Based Reasoning*, using 1 and 2 analogies, and some combinations of these techniques [10, 11]. Their results pointed out that *Linear Regression* and *Regression Tree* were better than *Case-Based Reasoning*, and in general *Regression Tree* provided the best results in terms of *MMRE* and *Pred*(0.25). Our results look in the same directions, in the context of Web applications, except for the application of *Case-Based Reasoning*. Indeed, while Briand *et al*. achieved weak results, the values of *MMRE*, *MdMRE*, and *Pred*(0.25) we obtained with *Case-Based Reasoning* suggest that further studies deserve to be carried out by applying this technique.

In the context of Web applications some studies have investigated techniques and/or measures for Web effort estimation [6, 19, 20, 37, 38, 39, 41, 44, 45, 46, 54, 57, 58], however of these only [19, 39] compared three or more techniques as in the present study. Other similarities can be found between [19, 39] and our study:

i. neither used costs drivers, in addition to size measures, i.e., the only independent variables employed were size measures;

ii. *Set1* represents a class of size measures common to the three studies (examples of these measures are number of Web pages, number of images/medias, number of internal link);

iii. the techniques *Stepwise Regression*, *Case-Based Reasoning*, and *Regression Tree*s were also used in the three studies;

iv. the studies looked at a common research question: which size measures are good indicators of Web application development effort and what estimation techniques can be considered effective to establish the relationships between the employed size measures and the development effort for the dataset?;

v. Prediction accuracy was measured in these empirical studies using the three common accuracy measures, namely *MMRE*, *MdMRE and Pred*(0.25).

However, the datasets used in these studies differed largely: Mendes *et al.* [39] used a dataset containing data on 37 Web hypermedia applications developed by postgraduate and MSc students attending a course at the University of Auckland (NZ), whereas in the present study and in [19] we used a dataset containing data on 15 Web applications developed by a single software company. Moreover, size measures and some of the techniques differed slightly. In particular, in [39] the *Web Objects* components were not considered. As for *Case-Based Reasoning* they considered three similarity measures and the best result was achieved by using weighted distance as similarity measure and 1 analogy. In the present empirical study the best result with *Case-Based Reasoning* was achieved using 2 analogies and inverse distance weighted mean as adaptation strategy (employing *Set2*). Mendes *et al.* [39] found that *Stepwise Regression* provided better performance than the other techniques when using measures of the type of *Set1*. Also in this study, *Stepwise Regression* presented good performance. However, the results obtained with *Stepwise Regression* using *Set1* were not significantly better than those obtained with *Case-Based Reasoning* (and *Regression Tree* combined with *Case-Based Reasoning*). This might be due to the use of the variable selection for *Case-Based Reasoning* that was not used in [39].

With respect to [19], in this paper we considered a different application of *Stepwise Regression* and *Case-Based Reasoning*. In particular:

i. we employed the manual selection of the variables for Stepwise Regression;

ii. we analyzed and compared two approaches to select the variables in the application of the Case-Based Reasoning technique, i.e., the Feature Subset Selection of ANGEL and the Pearson's Correlation test.

Differently from [19], the empirical analysis reported in this paper highlighted that Stepwise Regression provided positive results also with measures of Set1. This might be motivated by the application of the manual selection of the variables used in the present study. Moreover, the empirical results showed that Case-Based Reasoning with the Feature Subset Selection provided better results than those obtained in [19] without selecting variables, again suggesting that this feature can improve the performance of Case-Based Reasoning.

7 Conclusions

In this paper, we have reported on the results of an empirical study meant to compare some size measures and techniques. In particular, we focused on *Manual Stepwise Regression* (*MSWR*), *Case-Based Reasoning* (*CBR*), and combinations of *Regression Tree* with *CBR* (*RT+CBR*). For the application of *CBR* we analyzed two techniques for selecting variables, namely the *Feature Subset Selection* (*FSS*) of ANGEL and *Pearson's Correlation* test (*PC*). As predictors we employed the measures of set *Set1* and *Set2*, where *Set1* consists of some length measures that are specific of Web applications, while *Set2* consists of the components used to evaluate the *Web Objects* measure.

Although the study is based on an industrial dataset, it presents some limitations due to the number of Web applications in the dataset. Indeed 15 projects can be considered a quite small number from a statistical point of view to allow us to make generalization about the results obtained. However, some useful indications can be drawn that need to be validated in replicated studies. In particular, good results have been obtained using the employed size measures with *MSWR*, *CBR* with *FSS*, and *RT+CBR* with *FSS* (according to the Conte *et al.*'s thresholds [17]). Moreover, when using *MSWR*, measures in *Set1* gave significant better results than measures in *Set2*. As for *CBR* with *FSS* and *RT+CBR* with *FSS*, results did not show any significant differences among the employed size measures. Finally, *CBR* with *FSS* presented significant better results than *MSWR* and *CBR* with *PC* when using *Set2*. Furthermore, *RT+CBR* with *FSS* also presented significant better results than *MSWR* when using *Set2*. The comparison of the results reported in the present study with the ones of [19, 39] might suggest that the manual selection of the variables can improve the performance of *Stepwise Regression*, analogously the use of ANGEL's *FSS* can improve the performance of *CBR* (without variables selection).

What our empirical results suggest to practitioners of the software company is that they can use the measures in *Set1* if they employ *RT+CBR with FSS* or *MSWR*. Similarly, if they collect measures of *Set2* then they can employ *CBR* or *CBR* in combination with *RT*. Indeed, our analysis revealed that the results obtained using measures of *Set1* with *MSWR* and *RT+CBR* with *FSS* were significantly better than those obtained using mean or median of effort as estimated effort. On the other hand, using *Set2*

the results obtained applying *CBR* with *FSS* and *RT+CBR* with *FSS* were significantly better than those obtained using mean or median of effort as estimated effort.

It is worth noting that the results presented in this paper have provided some indications for the software company that supplied the data. They might be also relevant for other companies that develop projects similar to those used in our investigation. However, a replication of the study, in different setting and with a larger dataset, is required in order to generalize the results to other companies. Indeed, it is widely recognized that several investigations should be performed to verify/confirm empirical results [64]. Thus, as future work, we plan to collect and analyze data from other companies.

Acknowledgment

The authors would like to gratefully acknowledge the reviewers for their helpful comments and constructive criticisms.

References

1. A. Aamodt, E. Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches", *AI Communication*, IOS Press, 7(1), 1994, pp. 39-59.
2. S.M. Abrahão, O. Pastor, "Measuring the Functional Size of Web applications", in *International Journal of Web Engineering and Technology*, 1(1), 2003, pp. 5-16.
3. S.M. Abrahão, Geert Poels, O. Pastor, "Evaluating a Functional Size Measurement Method for Web Applications: An Empirical Analysis", in *Proceedings of Tenth International Software Metrics Symposium (METRICS'04)*, Chicago, Illinois, USA, 20004, pp. 358-369.
4. A. Abran, T. M. Khosgoftaar, A. Idri, "Fuzzy Analogy: A New Approach for Software Cost Estimation", in *Proceedings of the International Workshop on Software Measurement* (*IWSM'01*), Montreal, Canada, 2001, pp. 93-101.
5. A.J. Albrecht, "Measuring Application Development Productivity," in *Proceedings of the Joint SHARE/GUIDE/IBM Application Development Symposium*, Monterey, CA, 1979, pp. 83-92.
6. L. Baresi, S. Morasca, P: Paolini, "Estimating the Design Effort of Web Applications," *in Proc. of the 9th International Software Metrics Symposium*, Sydney, Australia, 2003, pp. 62-72.
7. V.R. Basili, L.C. Briand, W.L. Melo, "A Validation of Object-Oriented Design Metrics as Quality Indicators," *IEEE Transactions on Software Engineering*, 22(10), 1996, pp. 751-761.
8. V. R. Basili, F. Shull, F. Lanubile "Building Knowledge through Families of Experiments" *IEEE Transactions on Software Engineering*, 25(4), August 1999, pp. 456-473,.
9. B. Bohem, *Software Engineering Economics*, Prentice-Hall, Englewood Cloffs, NJ, 1981.
10. L. Briand, K. E. Emam, D. Surmann, I. Wiekzorek, K. Maxwell "An Assessment and Comparison of Common Software Cost Estimation Modeling Techniques," in *Proceedings of International Conference on Software Engineering,* (*ICSE'99*)*,* Los Angeles, USA, 1999.
11. L. Briand, T. Langley, I. Wiekzorek, "A Replicated Assessment and Comparison of Common Software Cost Modeling Techniques", *International Software Engineering Research Network Technical Report ISERN-99-15.*
12. L. Briand, I. Wieczorek. *Software Resource Estimation*. Encyclopedia of Software Engineering. Volume 2. P-Z (2nd ed.), Marciniak, John J. (ed.) New York: John Wiley & Sons, 2002, pp. 1160-1196.
13. L. Briand, J. Wüst, "Modeling Development Effort in Object-Oriented Systems Using Design Properties. *IEEE Transactions on Software Engineering* 27(11), 2001, pp. 963-986.
14. A. Comrey, "A First Course on Factor Analysis", London: Academic Press, 1973.

15. J. Cohen, *Statistical Power Analysis for the Behavioral Science*, Lawrence Erlbaum Hillsdale, New Jersey, 1988.
16. J. Conallen, *Building Web Applications with UML*, Addison-Wesley Object Technology Series, 1999.
17. D. Conte, H.E. Dunsmore, V.Y. Shen, *Software Engineering Metrics and Models*, The Benjamin/Cummings Publishing Company, Inc., 1986.
18. R.D. Cook, "Detection of influential observations in linear regression, *Technometrics*, 19, 1977, pp. 15-18.
19. G. Costagliola, S. Di Martino, F. Ferrucci, C. Gravino, G. Tortora, G. Vitiello, "Effort Estimation Modeling Techniques: A Case Study for Web Applications", in *ACM Proceedings of the 6th International Conference on Web Engineering (ICWE 2006)*, 2006, pp. 9-16.
20. G. Costagliola, F. Ferrucci, C. Gravino, G. Tortora, G. Vitello, "A COSMIC-FFP Based Method to Estimate Web Application Development Effort", in *LNCS* 3140, N. Koch, P. Fraternali, and M. Wirsing (Eds.): *International Conference on Web Engineering* (*ICWE'04*), Monaco, Germany, 2004, pp.161-165.
21. T. Dybå, N. B. Moe, E. M. Mikkelsen, "An Empirical Investigation on Factors Affecting Software Developer Acceptance and Utilization of Electronic Process Guides", in *Proceedings of IEEE International Symposium on Software Metrics* (METRICS'04), 2004, pp. 220-231.
22. T. Dybå, "An Empirical Investigation of the Key Factors for Success in Software Process Improvement", *IEEE Transactions on Software Engineering*, 31(5), 2005, pp. 410-424.
23. K.E. Emam, "A Primer on Object-Oriented Measurement", in *Proceedings of IEEE International Software Metrics Symposium* (*METRICS'01*), London, 2001, pp. 185-188.
24. R. L. Gorsuch, "*Factor Analysis*", . 2nd ed., Hillsdale: Lawrence Erlbaum Associates, 1983.
25. International Function Point Users Group: "Function Point Counting Practices Manual," Release 4.2.1, 2004.
26. G. Kadoba, M. Cartwright, L. Chen, M. Shepperd, "Experiences Using Case-Based Reasoning to Predict Software Project Effort", in *Proceedings of EASE 2000 Conference*, Keele, UK, 2000.
27. V, By Kampenes, T. Dybå, J. E. Hannay, D. I.K. Sjøberg, "A Systematic Review of Effect Size in Software Engineering Experiments", *Information and Software Technology* 4(11-12), 2007, pp.1073-1086.
28. B. A. Kitchenham, "A Procedure for Analyzing Unbalanced Datasets", *IEEE Transactions on Software Engineering*, 24(4), 1998, pp. 278-301.
29. A. Kitchenham, S. L. Pfleeger, L. M. Pickard "Case Studies for Method and Tool Evaluation", *IEEE Software* , 12(4), 1995, pp. 52-62.
30. A. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. El Emam, J. Rosenberg "Preliminary Guidelines for Empirical Research in Software Engineering", *IEEE Transactions on Software Engineering*, 28(8), 2002, pp. 721–734.
31. G. Kadoba, M. Shepperd, "Using Simulation to Evaluate Predictions Systems", in *Proceedings of International Software Metrics Symposium* (*METRICS'01*), England, 2001, pp. 349-358.
32. B. A. Kitchenham, L. M. Pickard, S. G. MacDonell, M. J. Shepperd, "What accuracy statistics really measure", *IEE Proceedings – Software*, 148(3), 2001, pp. 81-85.
33. S. Kunkel, U. Rosenqvist, R. Westerling, "The structure of quality systems is important to the process and outcome, an empirical study of 386 hospital departments in Sweden", BMC Health Serv Res. 7 (1), 2007, pp. 104.
34. R. Jeffery, J. Stathis, "Function point sizing: Structure, validity and applicability", *Empirical Software Engineering*, 1(1), 1996, pp. 11-30.
35. K. Maxwell, "Applied Statistics for Software Managers". Software Quality Institute Series, Prentice Hall, 2002.

36. Measurement in Software Engineering, Web site http://www2.umassd.edu/SWPI/ProcessBibliography/bib-measurement.html. Last visited on May 07, 2006.
37. E. Mendes, S. Counsell, N. Mosley, "Comparison of Web Size Measures for Predicting Web Design and Authoring Effort", *IEE Proceedings-Software* 149(3), 2002, pp. 86-92.
38. E. Mendes, S. Counsell, N. Mosley, "Web Metrics – Estimating Design and Authoring Effort", *IEEE Multimedia*, Special Issue on Web Engineering, 8(1), 2001, pp. 50-57.
39. 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.
40. E. Mendes, S. Counsell, N. Mosley, "Early Web Size Measures and Effort Prediction for Web Costimation", in *Proceedings of International Software Metrics Symposium* (*METRICS'03*), Sydney, Australia, 2003, pp. 18-39.
41. E. Mendes, S. Counsell, N. Mosley, "A Replicated Assessment of the Use of Adaptation Rules to Improve Web Cost Estimation", in *Proceedings of the International Symposium on Empirical Software Engineering* (*ISESE'03*), Rome, Italy, 2003, pp. 100.
42. E. Mendes, S. Di Martino, F. Ferrucci, C. Gravino. "Effort Estimation: How Valuable is it for a Web company to Use a Cross-company Data Set, Compared to Using Its Own Single-company Data Set?" In *ACM Proceedings of the 6th International World Wide Web Conference (WWW2007)*, Banff, Canada, 10-13 2007, pp 963 – 972.
43. E. Mendes, N. Mosley, "Towards a Taxonomy of Hypermedia and Web Application Size Metrics", in *Proceedings of International Conference of Web Engineering* (*ICWE 2005*), Sydney, Australia, 2005, pp. 110-123.
44. E. Mendes, B. Kitchenham, "A Comparison of Cross-company and Within-company Effort Estimation Models for Web Applications", in *Proceedings of Conference on Evaluation & Assessment in Software Engineering* (*EASE 2004*), Edinburgh, Scotland, 2004, pp. 47-55.
45. E. Mendes, B. Kitchenham, "Further Comparison of Cross-Company and Within-Company Effort Estimation Models for Web Applications", in *Proceedings of the International Symposium on Software Metrics* (*METRICS'04*), Chicago, IL, USA, 2004, pp. 348-357.
46. E. Mendes, N. Mosley, "Further Investigation into the Use of CBR and Stepwise Regression to Predict Development Effort for Web Hypermedia Applications", in *Proc. of International Symposium on Empirical Software Engineering* (*IESE'02*), Japan, 2002, pp. 79-90.
47. L. A. Merkle, C. S. Layne, J. J. Bloomberg, J. J. Zhang "Using factor analysis to identify neuromuscular synergies during treadmill walking", *Journal of Neuroscience Methods*, 82(2), 1998, pp. 207-214.
48. D. Montgomery, E. Peck, G. Vining, *Introduction to Linear Regression Analysis*, John Wiley & Sons, Inc., 3 Ed., 2001.
49. M. Morisio, I. Stamelos, V. Spahos, D. Romano, "Measuring Functionality and Productivity in Web-based applications: a Case Study", in *Proceedings of the International Software Metrics Symposium* (*METRICS'99*), Boca Raton, 1999, pp. 111-118.
50. M. Morisio, D. Romano, I. Stamelos, "Quality, Productivity, and Learning in Framework-Based Development: An Exploratory Case Study", in *IEEE Transactions on Software Engineering*, 28(9), 2002, pp. 876-888.
51. J. C. Munson, T. M. Khoshgoftaar, "The Detection of Fault-Prone Programs", *Transactions on Software Engineering* 18(5), 1992. pp. 423-433.
52. I. Myrtveit, E. Stensrud, "A Controlled Experiment to Assess the Benefits of Estimating with Analogy and Regression Models", *IEEE Transactions on Software Engineering*, 25(4), 1999, pp. 510-525.

53. D. Perry, A. Porter, L. Votta, "Empirical Studies of Software Engineering: A Roadmap", *The Future of Software Engineering*, Ed: Anthony Finkelstein, ACM Press, 2000, pp. 345-355.
54. D. Reifer, "Web-Development: Estimating Quick-Time-to-Market Software", *IEEE Software*, 17(8), 2000, pp. 57-64.
55. D. Reifer, "Web Objects Counting Conventions", *Reifer Consultants*, Mar. 2001. Available at: http://www.reifer.com/download.html.
56. T. Rollo, "Sizing E-Commerce", in *Proceedings of the ACOSM 2000 - Australian Conference on Software Measurement*, Sydney, 2000.
57. M. Ruhe, R. Jeffery, I. Wieczorek, "Cost Estimation for Web applications", in the *Proceedings of 25th International Conference on Software Engineering* (*ICSE'03*), Oregon USA, 2003, pp. 285 – 294.
58. M. Ruhe, R. Jeffery, I. Wieczorek, "Using Web Objects for Estimating Software Development Effort for Web Applications", in *Proceedings of International Software Metrics Symposium* (*METRICS'03*), Sydney, Australia, 2003, pp. 30-37.
59. M. Shepperd, C. Schofield, "Estimating Software Project Effort using Analogies", in *IEEE Transactions on Software Engineering*, 23(11), 2000, pp. 736-743.
60. M. Shepperd, C. Schofield, B. Kitchenham, "Effort Estimation using Analogy", in *Proceedings of International Conference on Software Engineering* (*ICSE'96*), Berlin Germany, 1996, pp.170-178.
61. E. Stensrud, T. Foss, B. Kitchenham, I. Myrtveit, "A Further Empirical Investigation of the Relationship between MRE and Project Size", *Empirical Software Engineering*, 8(2), 2003, pp. 139-161.
62. P. Umbers, G Miles, "Resource Estimation for Web Applications", in *Proceedings of tenth International Software Metrics Symposium* (*METRICS'04*), Chicago, Illinois USA, 2004, pp. 370-381.
63. F. Walkerden, R. Jeffery, "An Empirical Study of Analogy-based Software Effort Estimation", *Empirical Software Engineering*, 4(2), 1999, pp. 135-158.
64. C.Wohlin, P. Runeson, M. Host, M. C. Ohlsson, B. Regnell, A.Wesslen, *Experimentation in Software Engineering - An Introduction*. Kluwer, 2000.
65. M. V. Zelkowitz, D. R. Wallace, "Experimental Models for Validating Technology", *Computer*, 31(5), 1998, pp. 23-31.

Appendix

In this appendix, we report on the analysis carried out to verify the assumptions underlying the *Stepwise Regression*, to check the presence of possible outliers, to select the variables, and to analyze the fitting of the obtained model.

Regarding the assumptions required to carry out *Stepwise Regression*, we verified that residuals were independent and normally distributed; relationship between dependent and independent variables was linear. To this end, whenever the variables shown in Table 4 were highly skewed they were *transformed* before being used in *MSWR*. The employed transformation was the natural log (Ln), as suggested in other similar works [35]. A new variable containing the transformed values was created for each original variable that needed to be transformed. The new variables were identified as *Lvarname* (e.g. *LSSApp* represents the transformed variable *SSApp*). In addition, whenever a variable to be transformed had zero values, the natural logarithmic transformation was applied to the variable's value after adding 1.

To verify the stability of the effort model built using *MSWR*, the following steps were employed [44]:

- Use of a residual plot showing residuals vs. fitted values to investigate if the residuals were randomly and normally distributed.
- Calculate Cook's distance values [18] for all projects to identify influential data points. Any project with distance higher than 3 × (4/*n*), where *n* is the total number of projects, was removed from the data analysis [35]. Those with distances higher than 4/n but smaller than (3 × (4/n)) were removed in order to test the model stability, by observing the effect of their removal on the model. If the model coefficients remained stable and the *adjusted* R^2 improved, the highly influential projects were retained in the data analysis.

The application of the *MSWR* technique with *Set1* produced as best fitting model the one described in Table 15. The model's *adjusted* R^2 was 0.868, thus these variables explained 86.8% of the variation in *TotEff*.

The Equation of the final model's output was:

$$LTotEff = 4.358 + 0.508LSSApp + 0.192LIL + 0.241LMe \qquad (2)$$

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

$$TotEff = 78 \times SSApp^{0.508} \times IL^{0.192} \times Me^{0.241} \qquad (3)$$

Table 15: Prediction models obtained by using *MSWR* and Set1

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

The P-P plot (see Figure 4(a)) suggested that the residuals were normally distributed. The residual plot of Figure 4(b) revealed that one project presented a large residual. Since that project had Cook's distance between 4/15 and 3 x (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 presented similar values to those in the previous model. Thus, the data point was not removed from further analysis.

As for *Set2*, the best fitting model is described in Table 16. In this case, *MSWR* 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 explained 51.3% of the variation in *TotEff*. The Equation of the final model's output was:

$$LTotEff = 7.492 + 0.014EI \qquad (4)$$

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

$$TotEff = 1794 \times e^{0.048EI} \qquad (5)$$

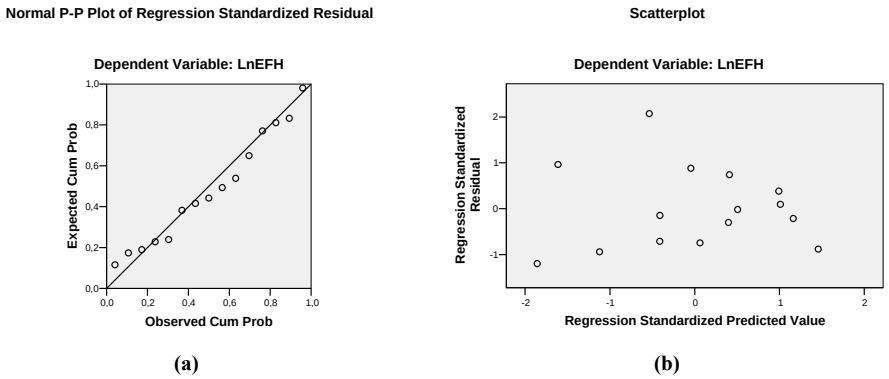

Figure 4 – P-P plot (a) and Residual plot (b) for the model obtained using Set1

Table 16 – Prediction models obtained by using *MSWR* and Set2

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

The P-P plot (see Figure 5(a)) suggested that the residuals were normally distributed while the residual plot presented in Figure 5(b)) showed one project with a large residual and this trend was also confirmed using Cook's distance. However, the analysis of the stability of the model suggested there was no need to remove the data point with a large residual for further analysis.

Summarizing, for both *Set1* and *Set2*, the values of *adjusted* R^2 suggested that the fit of the regression models obtained with *MSWR* was good. Moreover, the selected variables be considered significant indicators of efforts, as indicated by their *t*- and *p-values*.

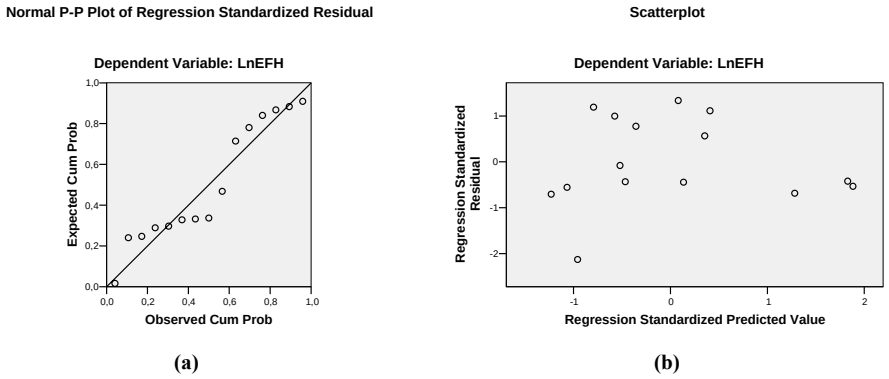

Figure 5 – P-P plot (a) and Residual (b) for the model obtained using Set2