

A Comparison of Size Measures for Predicting Web Design and Authoring Effort

Emilia Mendes
The University of Auckland
Computer Science
Department
Private Bag 92019 Auckland,
New Zealand
0064 9 3737599 ext. 6137
[emilia@cs.auckland.ac.
nz](mailto:emilia@cs.auckland.ac.nz)

Nile Mosley
MxM Technology POBox
3139
Shortland St. Auckland.
[nile_mosley@yahoo.co
m](mailto:nile_mosley@yahoo.com)

Steve Counsell
Birkbeck College, University
of London
Computer Science
Department
Senate House,
Russell Square
0044 020 7631 6700
[s.counsell@dcs.bbk.ac.
uk](mailto:s.counsell@dcs.bbk.ac.uk)

Abstract

Software practitioners recognise the importance of realistic estimates of effort to the successful management of software projects, the Web being no exception. Estimates are necessary throughout the whole development life cycle. They are fundamental when bidding for a contract or when determining a project's feasibility in terms of cost-benefit analysis. In addition, they allow project managers and development organisations to manage resources effectively.

Size, which can be described in terms of length, functionality and complexity, is often a major determinant of effort. Most effort prediction models to date concentrate on functional measures of size, although length and complexity are also essential aspects of size.

The first half of this paper describes a case study evaluation in which size metrics characterising length, complexity and functionality were obtained and used to generate effort prediction models for Web authoring and design. The second half describes the comparison of those size metrics as effort predictors by generating corresponding prediction models and comparing their accuracy using boxplots of the residuals. Results suggest that in general all categories presented a similar prediction accuracy.

1. Introduction

Software practitioners recognise the importance of realistic estimates of effort to the successful management of software projects, the Web being no exception. Prediction is a necessary part of an effective process [1], be it authoring, design, testing, or Web development as a whole.

This paper focuses on effort prediction for design and authoring processes. We adopt the classification proposed by Lowe and Hall [2] where authoring represents the creation of content and structure of the application and its presentation, and design covers the methods used for generating the structure and functionality of the application.

The data used to generate our prediction models was gathered through a quantitative case study evaluation where a set of proposed or reused [3-6] size metrics for effort prediction were measured. The Prediction models proposed were generated by statistical techniques, specifically Linear and Stepwise Regression.

Section 2 presents related work in development effort prediction for Web applications. Section 3 describes the case study evaluation and Section 4 presents and compares the prediction models using boxplots. Finally, we give our conclusions and comments on future work in Section 5.

2. Related Work in Web Engineering

To date there are only a few examples of effort prediction models for Web development in the literature as most work proposes methods and tools as a basis for process improvement and higher product quality [7-10].

Morisio et al. [11] describes an experiment involving the development of Web-based applications, using an object-oriented framework, where functional and code size metrics were collected. They applied an existing effort prediction model based on different reuse types and concluded that classical function points were not appropriate for the environment they used.

Rollo [12] explores the issues of counting a Web site in both IFPUG [13] and the MKII [14] methods. He discusses difficulties and contradictions between the rules and some approaches to Web site sizing. In addition, the same Web site is counted using the sizing method COSMIC-FFP¹ [15]. His conclusions

¹ COSMIC-FFP = Common Software Measurement International Consortium-Full Function Points

were that COSMIC proved to be the most flexible approach for counting the functional size of Web sites and can be applied to any Web site.

Mendes et al. [16] describe a case study where size and effort for Web applications, structured according to the Cognitive Flexibility [17], are measured and used to generate prediction models (linear and stepwise regression and analogy), which are then compared. Results showed that the best prediction was obtained using analogy.

Their contribution to the refinement of our case study is as follows: (i) consideration of reuse is fundamental when sizing Web applications (ii) COSMIC-FFP presented good results, which needed to be validated on other datasets (iii) confirmation that size measures are predictors of effort (iv) need for objective size measures collected automatically whenever possible.

3. Collecting Size Metrics for Web Authoring and Design Effort Prediction

3.1 Introduction

The case study evaluation measured Web applications size and their design and authoring effort. Four metrics representing confounding factors were also measured.

Our measurement activity was developed according to Fenton and Pfleeger's Conceptual Framework for Software Measurement [3], which is based on three principles:

1. Classifying the *entities* to be examined.
2. Determining relevant *measurement goals*.
3. Identifying the *level of maturity* the *organisation* has reached.

The classification of entities applied to the case study is presented in Table 1.

ENTITIES	ATTRIBUTES (Internal)
Products	
Web application	Page Count Media Count Program Count Total Page Allocation Total Media Allocation Total Embedded Code Length Reused Media Count Reused Program Count Total Reused Media Allocation Total Reused Code Length Connectivity Connectivity Density Total Page Complexity Cyclomatic Complexity Size _{CFSU} Structure
Processes	
Web authoring and design processes	Total Effort
Resources	
Developer	Authoring and Design Experience
Authoring tool	Tool Type

Table 1. Classification of Products, Processes and Resources for the case study

All metrics presented here are described in detail on Section 3.3.

Our measurement goals were documented using the Goal-Question-Metric (GQM) [18] (Table 2).

Goal	Question	Metric
Purpose: to measure Issue: Web design and authoring processes Object: process Viewpoint: developer's viewpoint	What attributes can characterise size of Web applications?	Page Count Media Count Program Count Total Page Allocation Total Media Allocation Total Embedded Code Length Reused Media Count Reused Program Count Total Reused Media Allocation Total Reused Code Length Connectivity Connectivity Density Total Page Complexity Cyclomatic Complexity Size _{CFSU} Structure
	What influence can developers have on the effort to design and author Web applications?	Authoring and design Experience
	How can design and authoring processes be measured?	Total Effort
	What influence can an authoring tool have on the effort required to author a Web application?	Tool Type

Table 2. The case study's goals, questions and metrics

Finally, the level of maturity within the Web application development community considered for our case study, measured according to the Capability Maturity Model (CMM) [19], is one.

3.2 The Case Study

The case study consisted of the design and authoring of Web applications aimed at teaching a chosen topic, structured according to the Cognitive Flexibility Theory (CFT) principles [17], using a minimum of 50 Web pages, which represents a medium-size Web application [2]. All subjects received training on the CFT authoring principles for approximately 150 minutes.

The relationship between the process model [2] used by the students to develop the applications and the steps to apply the CFT are presented in Figure 1:

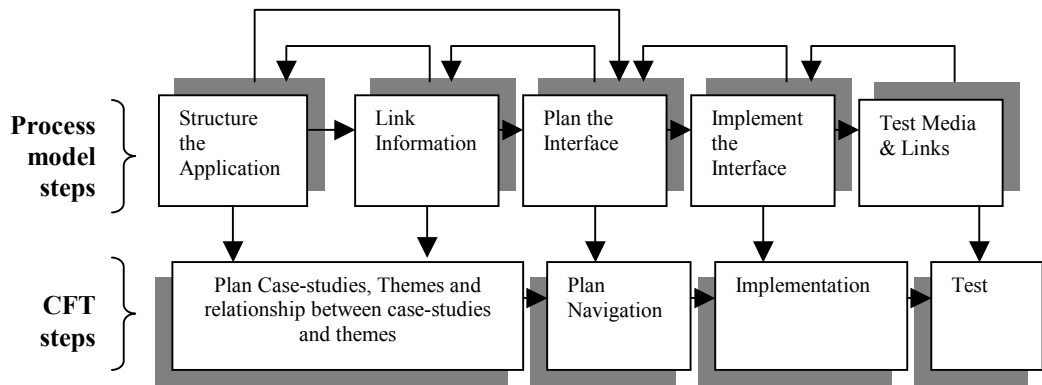


Figure 1 - Relationship between process model and CFT steps

Forty-three Computer Science students participated in the case study. Two questionnaires were used to collect the data. The first² asked subjects to rate their Web authoring experience using an ordinal scale, from no experience (one) to very good experience (five). The second questionnaire³ was used to measure size, confounding factors and design and authoring effort. On both questionnaires, each scale type was detailed to avoid misunderstandings.

To reduce learning effects, subjects were given a coursework prior to designing and authoring the Web applications, consisting of the creation of a Web site for the Matakoho Kauri Museum⁴, improving on their existing site.

Early analysis of the data showed that four questionnaires were missing fundamental information. They were not used in the data analysis. In addition, we identified three datapoints in the dataset for which the total effort to develop an application was noticeably high. Two of which did not present reasonable justification in the data (applications were small size and experience was not low). These were therefore removed from the dataset, leaving thirty-seven datapoints.

3.3 Metrics collected during the Case Study

The metrics collected during the case study evaluation represented attributes of three categories: Web application size, design and authoring effort and confounding factors [20]. Size metrics were organised in

² The questionnaire is available at <http://www.cs.auckland.ac.nz/~emilia/Assignments/exp-questionnaire.html>.

three tables (Tables 3-5) according to their categories: Length, Complexity, and Functionality. Effort, measured in person/hours, is presented in Table 6. Finally, confounding factors which could affect the internal validity of the experiment are presented in Table 7.

Metric	Description
Page Count *	Number of html or shtml files used in the Web application.
Media Count *	Number of non-reused media files used in the application.
Program Count	Number of non-reused cgi scripts [21], Javascript [22] files, Java applets [23] used in the application.
Total Page Allocation *	Total space (Mbytes) allocated for all the html or shtml pages used in the application.
Total Media Allocation *	Total space (Mbytes) allocated for all the media files used in the application.
Total Embedded Code Length	Total number of lines of code for all the programs used by an application.
Reused Media Count *	Number of reused/modified media files.
Reused Program Count	Number of reused/modified programs.
Total Reused Media Allocation *	Total space (Mbytes) allocated for all the reused media files used in the application.
Total Reused Code Length	Total number of lines of code for all the programs reused by an application.

Table 3. Length Metrics

Media and program files are considered reused only if they already existed externally to the Web application and are copied for use (as a black box or not) within the application.

³ The questionnaire is available at <http://www.cs.auckland.ac.nz/~emilia/Assignments/questionnaire.html>.

⁴ <http://www.hmu.auckland.ac.nz:8001/gilchrist/matakohe/>.

* Suitable for static Web applications, where the number of dynamically generated links and/or pages is absent.

Metric	Description
Connectivity *	Total number of internal links ⁵ . We do not include dynamically generated links.
Connectivity Density [3] *	Connectivity / Page Count.
Total Page Complexity *	$\sum_1^{PageCount} \text{Numb. diff. types media in a page} / \text{PageCount}$
Cyclomatic Complexity [3] *	(Connectivity - Page Count) + 2.

Table 4. Complexity Metrics

Metric	Description
Size _{CFSU} [15]	$\Sigma \text{size(Entries)} + \Sigma \text{size(Exits)} + \Sigma \text{size(Reads)} + \Sigma \text{size(Writes)}$

Table 5. Functionality Metric

Metric	Description
Total Effort	Estimated elapsed time (number of hours) it took a subject to design and author the application. It is the summation of the elapsed times spent on each process model's step.

Table 6. Effort Metric

⁵ Subjects did not use external links to other Web sites. All the links pointed to pages within the application only.

Metric	Description
Tool Type	Measures the type of tool used to author/design the Web pages: WYSIWYG (What You See Is What You Get), semi-WYSIWYG or text-based.
Authoring and Design Experience	Measures the authoring/design experience of a subject.
Structure ⁶ *	Measures how the main structure (backbone) of the application is organised.

Table 7. Confounding Factors

Compactness and Stratum [25] were not considered in this evaluation as they could only be measured subjectively, compromising the validity of the results.

3.4 Measuring Size metrics

Complexity and Length metrics were collected using a questionnaire and three spreadsheets which subjects were required to complete. All complexity and length metrics, except reuse, were re-measured by using the applications developed.

Functionality was measured using the COSMIC-FFP [15] measurement method. We counted the number of Cosmic Functional Size Units (CFSUs) in the Web application. The COSMIC-FFP method is a standardised measure of software. It is applicable to domains such as application software (banking, insurance, accounting etc), real-time software, hybrids of both. We have adapted the method to Web applications, in a way similar to [12].

⁶ The structure can be a sequence, hierarchy or network. A sequential structure corresponds to documents linearly linked; a hierarchical structure denotes documents linked in a tree shape and a network structure for documents linked in a net shape [24].

The COSMIC-FFP method involves applying a set of rules and procedures to a given piece of software, as perceived from the perspective of its Functional User Requirements (FURs)⁷. The result is a numerical "value of quantity"⁸ representing the functional size of the software. The model consists of two parts: the context model and the software model.

The context model helps establish what is part of the software and what is part of the software's operating environment. It identifies the software boundaries and the functional flow of data attributes. This flow has two directions: front-end and back-end, representing respectively ENTRIES & EXITS and READS & WRITES (See Figure 2).

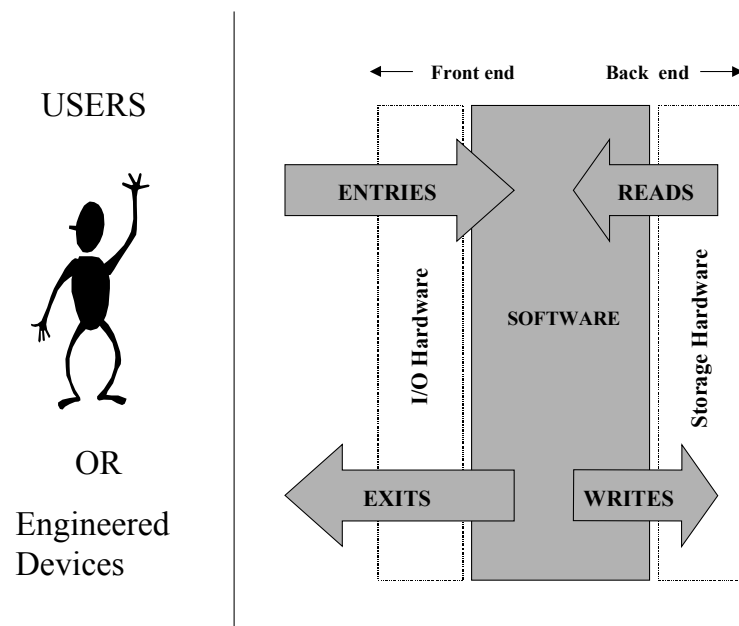


Figure 2 - Generic flow of data attributes from a functional perspective [15]

The software model assumes that the software to be mapped and measured manipulates pieces of information designated as data groups, which consist of data attributes. The software functional requirements are implemented by a set of functional processes, each is an ordered set of sub-processes performing either data movement or data manipulation (see Figure 3).

⁷ The term Functional User Requirement is an ISO expression designating a subset of the user requirements. It represents the user practices and procedures that the software must perform to fulfil the user's needs, excluding Quality requirements and any technical requirements [ISO/IEC 14143:1997].

⁸ The value of a quantity is the magnitude of that particular quantity, generally expressed as a unit of measurement multiplied by a number [IOS-1993].

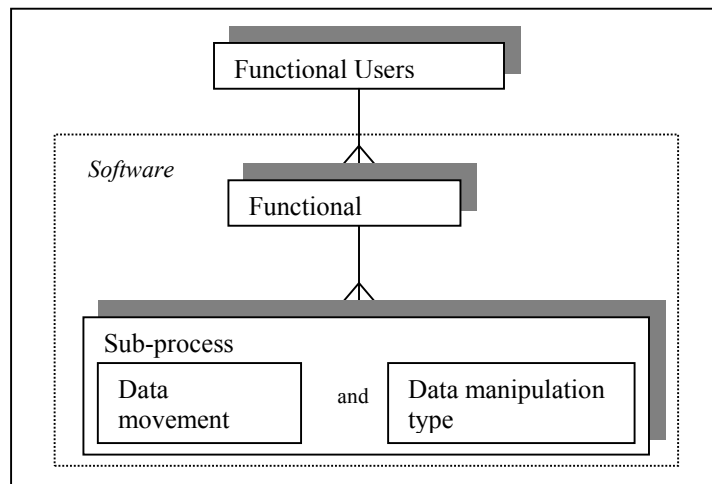


Figure 3 - A general software model for measuring functional size [15]

The functional size of software is directly proportional to the number of its data movement sub-processes. The unit of measure used is 1 CFSU, equivalent to a single data movement at the sub-process level. Consequently, the total size is given by the number of Entries + number of Exits + number of Reads + number of Writes [15].

The context model for Web applications in the case study is presented in Figure 4.

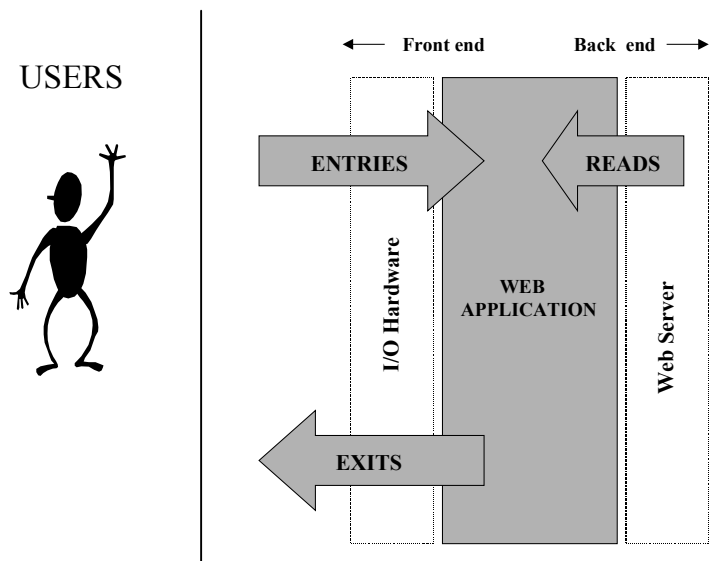


Figure 4 - Flow of data attributes through Web application from a functional perspective

The counting rules used corresponding to each entry, exit, read and write are as follows:

- Each "HREF" tag counted as 1 entry + 1 read + 1 exit. By pressing a link, the user sends an entry to the application, which reads the data from the Web server and shows the contents to the user. We assumed the Web server to be the Data Storage Hardware. Although several subjects used in-line images [21], they were not considered as a separate data group since every "IMG SRC" tag will be downloaded automatically with the Web page. The subjects who participated in this case study did not use external resources (images, sound and video), which use the "HREF" tag and have to be downloaded from the Web server upon request.
- In relation to the types of programs used, subjects used either Java applets or Javascript. For each Java applet we counted one entry + one exit (one entry to activate the applet and one exit showing the applet). As applets are also downloaded from the server at the same time as the Web page, we did not consider a "read" flow of data.
- For each Javascript file we only considered the "HREF" tag ("entry" flow of data) to be the source of interaction between user and application.

We did not have a Functional Users Requirements documents for the Web applications, so we counted the number of CFSUs for each application using the implementation provided by the subjects only. Some links ("HREF" tags) existed to give readers more information about the contents of the application; others to give better navigation guidance (Site map, back to main page, previous page, next page etc). We understand that both types of links are equally important in achieving the purpose of conveying information in a meaningful way and a reasonable level of functionality. We believe that our counts would map easily to any document in which navigation and linking topologies were represented.

3.5 Validity of the Case Study

Comments on the validity of the case study are as follows:

- There are difficulties measuring Web applications because: (i) design and authoring effort are difficult concepts to measure. (ii) there is no standard on what constitutes the design and authoring processes of a Web application. Therefore, the activities used in this study may not be representative of all practices employed in the Web community.
- The metrics collected, apart from effort, experience and structure, were all objective and precisely quantifiable. The granularity of effort data was made consistent with the Web authoring process employed. In addition, the scale used to measure experience and the three types of structure of an application were described in detail in both questionnaires.
- There were four confounding factors in the case study evaluation:
 - ❖ Subjects' authoring and designing experience.
 - ❖ Maturation effects, i.e. learning effects caused by subjects learning as an experiment proceeds.
 - ❖ Structure of the application.
 - ❖ Tools used to help author and design the Web application.

In relation to those confounding factors, the data collected revealed that:

- ❖ Subjects' authoring and design experiences were mostly scaled little (experience=2) or average (experience=3), with a low skill differential⁹. A scatterplot between experience and effort was used to investigate their relationship. Most datapoints fell within clusters between the intervals 10 to 80 hours, either for experience 2 or 3. Consequently the original dataset was left intact.
 - ❖ Subjects had to develop a Web application as part of their first assignment. They also received training in the CFT principles, reducing maturation effects.
 - ❖ Notepad (or similar text editor) and FirstPage were the two tools most frequently used. Although they differ with respect to the functionality offered, a scatterplot between total-effort and tool revealed that for both tools most datapoints fell within the interval between 10 to 80 hours. Consequently, confounding effects from the tools were reduced.
- The selection effects, which are due to natural variations in subject performance, were reduced, as individual differences were spread across all applications.
 - The instrumentation effects in general did not occur in this evaluation; the questionnaires used were the same.
 - The results may be domain dependent as all subjects answered the questionnaires based on their experience in developing Web applications for education. This evaluation should therefore be repeated in domains other than education if the results are to be generalised to other domains.
 - The applications did not have more than 103 documents (mean=57) and 1501 links (mean 616). These numbers represent medium size applications, according to [2]. Consequently they might not be representative of the large Web applications for education developed by some organisations. However, the applications developed had similar to or better interface and contents quality than Web applications developed by professionals (as observed by one of the authors when marking the applications), suggesting that the results of the case study are likely to scale-up.

⁹ low difference between skill levels

- A threat to subject generalisability [26] may exist when the subject population is not drawn from a wider population. The subjects that participated in the case study were either final year undergraduate students or MSc students. It is likely that they present skill sets similar to Web professionals at the start of their careers. The use of students as subjects, while sometimes considered unrealistic, is justified for two reasons: firstly, empirical evidence by Boehm-Davis and Ross [27] indicates that students are equal to professionals in many quantifiable measures, including their approach to developing software; secondly, for pragmatic considerations, having students as subjects was the only viable option for this case study.

4. Estimation Process

4.1 Introduction

For the collected data, both linear and stepwise regressions [28,29] were used to generate effort prediction models for Web design and authoring, which were then compared according to their prediction accuracy. Stepwise regression has been frequently used as a benchmark [29-33] and is regarded by some as a good prediction technique [34].

When applying linear regression, whenever we had more than one predictor variable to generate a model, we used all variables simultaneously.

Both linear regression and stepwise regression were calculated using SPSS 10.0.5.

4.2 Results using Linear and Stepwise Regression

Predictors and response variable had a linear relationship. Before generating the prediction models we looked for statistically significant correlation between predictors in order to avoid spurious relationships. In addition, we generated plots of the residuals, which revealed distributions that did not present any disturbing patterns.

The formulas obtained by using linear regression and stepwise regressions are as follows (see Table 8):

Size Metrics	Linear Regression	Adj. R ²	Stepwise Regression	Adj. R ²
Length	-3.158 + 0.558 * Page Count + 0.03937 * Total Page Allocation + 0.02795 * Total Reused Code Length	0.421	2.963 + 0.03374 * Total Reused Code Length + 0.506 * Page Count	0.418
Complexity	20.911 + 0.0280 * Connectivity	0.310	17.840 + 0.02703 * Connectivity	0.355
Functionality	20.061 + 0.00895 * Size _{CFSU}	0.378	20.061 + 0.00895 * Size _{CFSU}	0.378

Table 8 - Prediction Models using Linear and Stepwise Regression

The metrics used to generate the prediction models were measured on a ratio scale.

In relation to Length, both prediction models included *Page Count* and *Total Reused Code Length*. We did not assume that either *Page Count* or *Total Page Allocation* would have a spurious relationship with effort since they did not have a statistically significant correlation and in practice it is quite possible to have Web applications with small number of pages and using a good amount of storage space. The opposite is likely to happen as well, in particular when the application is more text-based than graphics-based.

In relation to Complexity, we only used *Connectivity* and *Total Page Complexity* in the regression models, as there were strong statistically significant correlations between *Connectivity* and *Connectivity Density*; *Connectivity* and *Cyclomatic complexity*. *Connectivity* presented the best statistical correlation with effort, when compared to *Connectivity Density* and *Cyclomatic Complexity*. As shown in Table 7, both regression models for Complexity included *Connectivity* only.

In relation to functionality, both prediction models included *Size_{CFSU}*.

Unfortunately, no model presented reasonable prediction, based on their adjusted-R². That indicates that replications of our case study are clearly necessary to investigate whether our size measures are inadequate or some possible bias in the manual data collection influenced our results.

4.3 Comparing the Prediction Models

We used boxplots of the residuals to compare the two prediction models generated, for each one of the Size categories, as suggested by Pickard et al. [35] and Kitchenham et al. [36]. It gives a good indication of the distribution of the residuals and can help explain the behaviour of the summary statistics [37].

The boxplots of the residuals (difference between the actual and predicted values for effort), presented in Figure 5, were used to compare:

- The range of the residuals from each of the different models, for each size category.
- The distribution of the residuals.
- The number of outliers.

The Legend used in Figure 5 is as follows:

LENGTHL - Residuals for Length based on estimated effort generated using a linear regression model
LENGTHS - Residuals for Length based on estimated effort generated using a stepwise regression model
COMPLL - Residuals for Complexity based on estimated effort generated using a linear regression model
COMPLS - Residuals for Complexity based on estimated effort generated using a stepwise regression model
FUNCLS - Residuals for functionality based on estimated effort generated using linear and stepwise regression models. As both revealed the same results, only one box was used.

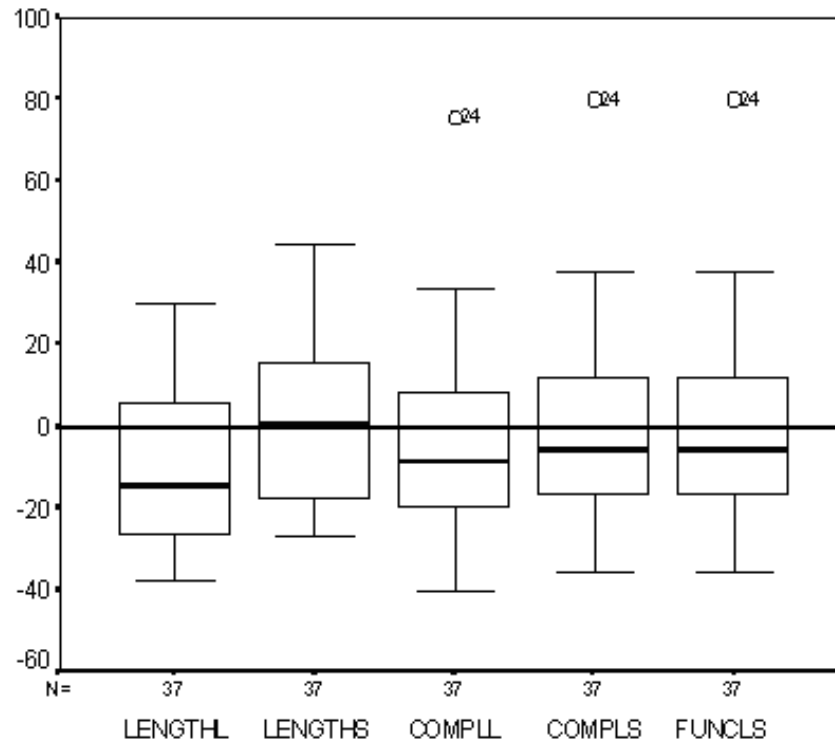


Figure 5 - Boxplot of residuals for each prediction system used, organised by size category

The boxplots obtained indicate that:

- The distributions of COMPLL, COMPLS and FUNCLS have a slight right skew.
- The distributions for LENGTHL and LENGTHS are positively and negatively skewed respectively.
- COMPLL, COMPLS and FUNCLS presented one outlier each, which is the same datapoint. Removing the outlier from the dataset did not influence the equations produced.
- LENGTHL and LENGTHS have a slightly tighter spread than COMPLL, COMPLS and FUNCLS, being less variable and less peaked.
- Except for LENGTHS, all medians represented values below zero, meaning that the estimates were biased towards over-estimation.

- For LENGTHL, COMPLL, COMPLS and FUNCLS more than 50% of datapoints are negative (below zero) indicating estimated effort greater than actual effort (over-estimation).
- All distributions seem to be flatter as the size of the boxes are not small when compared to the tails.
- None of the models produced reasonable accurate estimates of effort.

We also tested the statistical significance of the absolute residuals for all size categories to test whether the differences exist caused by chance alone or are legitimate.

The statistical significance of the results was tested using a T-test of mean difference between paired residuals and a Mann-Whitney U Test, similar to [38]. However, as the data used is skewed, preference should be given to the results presented using the Mann-Whitney U Test as it is a non-parametric test.

The results obtained are as follows (Table 9):

T-test						Mann-Whitney U Test					
Linear Regression			Stepwise Regression			Linear Regression			Stepwise Regression		
Len.	Com.	N	Len.	Com.	N	Len.	Com.	N	Len.	Com.	N
Len.	Fun.	N	Len.	Fun.	N	Len.	Fun.	N	Len.	Fun.	N
Com.	Fun.	N	Com.	Fun.	N	Com.	Fun.	N	Com.	Fun.	N
Com. = Complexity Fun. = Functionality Len. = Length											
N = no statistical significance Y = statistical significance											

Table 9 - Statistical significance for the absolute residuals

No statistical significance was found, indicating that the models did not produce significantly different residual values. Consequently, we can conclude that there was no single technique that produced a better fitting model than the others.

5. Conclusions and Future Work

The first half of this paper presented a case study evaluation in which we measured attributes of Web applications corresponding to three size categories, namely Length, Complexity and Functionality. The second half of the paper suggested effort prediction models generated for each one of the size categories, using two statistical techniques, Linear regression and Stepwise Multiple Regression. These models were compared using boxplots of the residuals and none presented reasonable accurate estimates of effort. In addition, there was no statistical significance of the absolute residuals, which means that there was no single technique to produce a better fitting model than the others.

At least for the Web applications used in our dataset it seems that measuring the functionality of those applications was very much related to the number of links that the application has. As we counted the number of Cosmic Functional Size Units using the final implementation, rather than the Functional User Requirement model, further investigation is clearly necessary to determine whether or not functionality is highly correlated to connectivity.

Our future work includes using the COSMIC-FFP model to measure the functionality of several types of static and dynamic Web sites and to use real datasets from industrial practice.

To conclude, there is an urgent need for adequate Web development effort prediction at an early stage in the development. As the use of the Web as a resource delivery environment increases, effort estimation can contribute significantly to the reduction of costs and time involved in developing Web applications.

6. References

- [1] PFLEEGER, S. L., JEFFERY, R., CURTIS, B., AND KITCHENHAM, B.: 'Status Report on Software Measurement', *IEEE Software*, March/April, 1997.
- [2] LOWE, D., AND HALL, W.: 'Hypertext and the Web - An Engineering Approach', (John Wiley & Sons Ltd., eds.), 1998.
- [3] FENTON, N. E., AND PFLEEGER, S. L.: 'Software Metrics, A Rigorous & Practical Approach', 2nd edition, (PWS Publishing Company and International Thomson Computer Press), 1997.
- [4] HATZIMANIKATIS, E., TSALIDIS, C. T. AND CHRISTODOULAKIS, D.: 'Measuring the Readability and Maintainability of Hyperdocuments', *J. of Software Maintenance, Research and Practice*, 1995, 7, pp. 77-90.

- [5] WARREN, P., BOLDYREFF, C., MUNRO, M.: 'The Evolution of Websites', Proc. Seventh International Workshop on Program Comprehension, IEEE Computer Society Press, Los Alamitos, Calif., 1999, pp. 178-185.
- [6] MCDONELL, S. G., AND FLETCHER, T.: 'Metric Selection for Effort Assessment in Multimedia Systems Development', Proc. Metrics'98, 1998.
- [7] GARZOTTO, F., PAOLINI, P., AND SCHWABE, D.: 'HMD – A Model-Based Approach to Hypertext Application Design', *ACM Transactions on Information Systems*, 1993, **11**, (1), January.
- [8] SCHWABE, D. AND ROSSI, G.: 'From Domain Models to Hypermedia Applications: An Object-Oriented Approach', Proceedings of the International Workshop on Methodologies for Designing and Developing Hypermedia Applications, Edimburgh, September, 1994.
- [9] BALASUBRAMANIAN, V., ISAKOWITZ, T., AND STOHR, E. A.: 'RMM: A Methodology for Structured Hypermedia Design', *Communications of the ACM*, 1995, **38**, (8), August.
- [10] CODA, F., GHEZZI, C., VIGNA, G., AND GARZOTTO, F.: 'Towards a Software Engineering Approach to Web Site Development', Proceedings of the 9th International Workshop on Software Specification and Design, 1998, pp. 8-17.
- [11] MORISIO, M., STAMELOS, I., SPAHOS, V. AND ROMANO, D.: 'Measuring Functionality and Productivity in Web-based applications: a Case Study', Proceedings of the Sixth International Software Metrics Symposium, 1999, pp. 111-118.
- [12] ROLLO, T.: 'Sizing E-Commerce', Proceedings of the ACOSM 2000 - Australian Conference on Software Measurement, Sydney, 2000.
- [13] IFPUG: Function Point Counting Practices Manual, Release 4.0, International Function Point Users Group, Westerville, Ohio, 1994.
- [14] UNITED KINGDOM SOFTWARE METRIC ASSOCIATION: 'MKII Function Point Analysis Counting Practices Manual', version 1.3.1, September, 1998, 92 pages.
- [15] COSMIC: 'COSMIC-FFP Measurement manual', version 2.0, <http://www.cosmicon.com>, 1999.
- [16] MENDES, E., COUNSELL, S., AND MOSLEY, N.: 'Measurement and Effort Prediction of Web Applications', Proc. Second ICSE Workshop on Web Engineering, 4 and 5 June 2000; Limerick, Ireland, 2000.

- [17] SPIRO, R. J., FELTOVICH, P. J., JACOBSON, M. J., AND COULSON, R. L.: 'Cognitive Flexibility, Constructivism, and Hypertext: Random Access Instruction for Advanced Knowledge Acquisition in Ill-Structured Domains', In: L. Steffe & J. Gale, eds., 'Constructivism', (Hillsdale, N.J.:Erlbaum), 1995.
- [18] BASILI, V., CALDIERA, G., AND ROMBACH, D.: 'The Goal Question Metric Approach.' In: Encyclopaedia of Software Engineering, Wiley, 1994.
- [19] PAULK, M. C., CURTIS, B., CHRISSIS, M. B., AND WEBER, C. V., "Capability Maturity Model, Version 1.1", *IEEE Software*, 1993, **10**, (4), July, pp.18-27.
- [20] KITCHENHAM, B., PICKARD, L., PFLEEGER, S. L.: 'Case Studies for Method and Tool Evaluation', *IEEE Software*, 1995, July, pp. 52-62.
- [21] ABERNETHY, K., ALLEN, T.: 'Exploring the Digital Domain', (Brooks/Cole Publishing Company), 1999.
- [22] FLANAGAN, D.: 'JavaScript – The Definitive Guide', Second edition, (O'Reilly & Ass.), January, 1997.
- [23] GOSLING, J. AND MCGILTON, H.: 'The Java Language Environment: a White Paper', Technical Report, Sun Microsystems, October, 1995.
- [24] WHALLEY, P.: 'Models of Hypertext Structure and Learning', In: D. H. Jonassen and H. Mandl, eds. Designing Hypermedia for Learning, (Berlin, Heidelberg: Springer-Verlag), 1990, pp. 61-67.
- [25] BOTAFOGO, R. A., RIVLIN, E., AND SHNEIDERMAN, B.: 'Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics', *ACM TOIS*, 1992, **10**, (2), pp. 143-179.
- [26] PORTER, A, SIY, H. P., TOMAN, C. A., AND VOTTA, L. G.: 'An Experiment to Assess the Cost-Benefits of Code Inspections in Large Scale Software Development', *IEEE Transactions of Software Engineering*, 1997, **23**, (6), pp. 329-346.
- [27] BOEHM-DAVIS, D. A., AND ROSS, L. S.: 'Program design methodologies and the software development process', *International Journal of Man-Machine Studies*, 1992, **36**, pp. 1-19, Academic Press Limited.
- [28] MASON, R. L., GUNST, R. F., AND HESS, J. L.: 'Statistical Design and Analysis of Experiments with applications to Engineering and Science', (Wiley:New York, Chichester, Brisbane, Toronto, Singapore), 1989, pp. 435-440.

- [29] SCHROEDER, L., SJOQUIST, D., AND STEPHAN, P.: Understanding Regression Analysis: An Introductory Guide. No 57. In Series: Quantitative Applications in the Social Sciences, (Sage Publications, Newbury Park, CA, USA), 1986.
- [30] SHEPPERD, M.J., SCHOFIELD, C., AND KITCHENHAM, B.: 'Effort Estimation Using Analogy', Proc. ICSE-18, IEEE Computer Society Press, Berlin, 1996.
- [31] KADODA, G., CARTWRIGHT, M., AND SHEPPERD, M.J.: Issues on the effective use of CBR technology for software project prediction, Proceedings of the 4th International Conference on Case-Based Reasoning, ICCBR 2001, Vancouver, Canada, July/August, pp: 276-290.
- [32] SHEPPERD, M.J., AND KADODA, G.: Using Simulation to Evaluate Prediction Techniques, Proceedings of the IEEE 7th International Software Metrics Symposium, London, UK, 2001, pp: 349-358.
- [33] MENDES, E., MOSLEY, N., AND COUNSELL, S.: Web Metrics – Estimating Design and Authoring Effort. *IEEE Multimedia*, Special Issue on Web Engineering, January-March, 2001, pp.50-57.
- [34] KOK, P., KITCHENHAM, B. A., KIRAKOWSKI, J.: The MERMAID Approach to software cost estimation, ESPRIT Annual Conference, Brussels, 1990, pp: 296-314.
- [35] PICKARD, L. M., KITCHENHAM, B.A., AND LINKMAN, S.J.: 'An investigation of analysis techniques for software datasets', In: Proc. Sixth International Symposium on Software Metrics, IEEE Computer Society Press, Los Alamitos, CA, 1999.
- [36] KITCHENHAM, B.A., MACDONELL, S.G., PICKARD, L.M., SHEPPERD, M.J.: 'What accuracy statistics really measure', *IEE Proc. Software*, 2001, **148**, (3), June, pp. 81-85.
- [37] BUSINESS STATISTICS,
<http://gsbwww.uchicago.edu/fac/michael.parzen/teaching/opre504.html#rbp>
- [38] MYRTVEIT, I, AND STENSRUD, E.: 'A Controlled Experiment to Assess the Benefits of Estimating with Analogy and Regression Models', *IEEE Transactions on Software Engineering*, 1999, **25**, (4), Jul./Aug., pp. 510-525.