Where are we? A Review of Effort and Cost Estimation Techniques used for Web Applications

Lenandlar Singh[1]
[1]Lecturer, Department of Computer Science, University of Guyana, Guyana, South America

Abstract

The rapid growth of the World Wide Web(WWW) and the growing demand for web sites and web applications to compliment and even replace business processes have witnessed an equally rapid growth in the number of companies offering web development solutions.

A reliable effort and cost estimation approach can help companies bid and compete for web development projects and more importantly, assist in timely and within budget development of projects undertaken.

This paper examines the literature on Effort and Cost Estimation Techniques for both Web Applications and Traditional Software Applications and the effort and cost estimation techniques used. It attempts to argue that web applications have inherent differences compared to traditional software applications to merit customised estimation approaches from traditional software effort estimation through. However, we suggest some areas where web applications may have converging characteristics to traditional software applications and that the techniques used currently in traditional software estimation may be useful to explore for web application estimation. This paper concludes by suggesting some recommendations for consideration.

Keywords: *Effort Estimation, Cost Estimation, Web Applications, Review*

1. Introduction

Earliest evidence on research conducted to derive effort and cost estimations required to develop software applications dates back to the early 1960s [1]. Ever since there have been a proliferation of methods and techniques, together with a wide range of technologies, processes, experienced software developers and the range of medium through which these applications are deployed. The World Wide Web is one such new invention.

Over the last 10 years the WWW has witnessed an enormous growth in the amount, size and type of web sites and web applications available to users worldwide. This growth coincides with the number of users and customers demand for such resources, the rapid growth and increasing capability of technologies available and a growing skills base available to develop the websites and web applications to supply this growing need. Additionally, there is a rapid increase in the number of companies that are providing web sites and web application development services.

[1]http://www.cs.auckland.ac.nz/compsci702s1c/lectures/emilia/chapter%201%20final.pdf

A reliable effort and cost estimation method and strategy are important for companies in general and project managers specifically to bid for and complete projects with time and budget [2] points out that an ‘effort estimation is a critical activity for planning and monitoring software project development and for delivering the project within time and budget’. They further state that a ‘reliable effort estimation is crucial for a successful web application development planning. This is no different for web application development as suggested by [3]. A number of different approaches have been used over the years and many different tools were developed to help the estimation process.

There are possibly many significant reasons why effort and cost estimation be undertaken by a company. A report published by the Cost Cutter Consortium in 2000 as cited in Mendes, Mosley & Counsell [1] shows some alarming statistics resulting from large web-based projects that were outsourced:

- 84% of surveyed delivered projects did not meet business needs.
- 53% of surveyed delivered projects did not provide the required functionality.
- 79% of surveyed projects presented schedule delays.
- 63% of surveyed projects exceeded their budget.

Scheduling delays and exceeding budgets are two of the issues that may be directly or indirectly impacted by the effort and cost estimation techniques used by these companies.

The rest of this paper is organised as follows: **Section 2** provides background and a general overview of the nature of web applications, discussed traditional Software Engineering estimation techniques. **Section 3** surveys the work done in effort estimation for web applications to show how and when traditional software approaches are adopted. **Section 4** discusses the potential of the research done and its value and possible implications for tailoring traditional approaches to fit local needs and how this may affect adoption of research results. Finally, **Section 5** gives conclusion and suggestions for future work.

2. Background & Nature of Web Applications

Web Applications Effort and Cost Estimation is a complex task due to a number of different but compelling reasons compared to traditional software engineering effort and cost estimation techniques. [4] listed a number of factors that makes web application development slightly different from traditional web application development, indicates that there is ‘no standard to sizing web applications’ given that there is a wide and diverse set of technologies that can be used to develop web applications e.g. Java (Servlets, Java Beans, Applets, Java Server Pages), HTML, JavaScript, XML, XSL, PHP, ASP.NET etc.

Many different metrics have been used to estimate the size and therefore the effort required to complete a web application project. Examples include the number of web pages, the number of multimedia elements, and the number of links and so on. Additionally, there have been attempts to ‘apply Function Point principles to sizing web applications [5][3]. This approach seeks to derive from a combination of size metrics, a number of functional requirements that would be needed in order to develop the application.

Additional reasons includes: ‘web Development Processes differ substantially traditional approaches [software engineering]’**, [5][**6] as cited in[3]. [7][8] as cited in[3] notes that ‘people involved in web development are represented by less experienced [than traditional software] programmers, users as developers, graphic designers and new hires straight from university’. To add to this is the fact that traditionally, web applications ‘primary goal is to bring quality applications to market as quickly as possible, varying from a few weeks to six months’, [9] [6][10] .

Another important reason why it is often difficult to estimate effort and cost estimation for web applications is due to the fact that the ‘processes employed are generally ad hoc [7]as cited in [3] when compared to more established approached employed in software engineering and development. However, there are signs that this is changing for example [11] investigated the use of agile methods for web application development.

[12] have grouped and summarized the differences between Web and Software Development into 12 areas as follows:

1. Application Characteristics
2. Primary Technologies
3. Approach to Quality Delivered
4. Development Process Drivers
5. Availability of the Application
6. Customer (Stakeholders)
7. Update Rate (Maintenance Cycles)
8. People Involved in Development
9. Architecture and Network
10. Disciplines Involved
11. Legal, Social and Ethical Issues
12. Information Structuring and Design

It is under this belief that the Web Engineering Community is trying to develop and adapt techniques already existing in Software Engineering to meet Web Development needs.

2.1. Traditional Estimation Techniques

Many different effort and cost estimation techniques have been 'proposed and used over the last 30 years [in the field of Software Engineering],[3], Ruhe et al [1]. They fall into three (3) general categories, [13].

1) **Expert Judgement (EJ)** – technique that uses experts' opinion and previous experiences to derive an estimate. These methods are not very explicit and therefore not very repeatable, [3]. [14] as cited in[3] notes that 'judgement can be effective estimation tool on its own or as an adjusting factor for algorithmic models', even though they are not easily quantifiable[3].

2) **Algorithmic Models (AM)** – 'most popular technique' according to [3] and 'attempts to represent relationships between effort and one or more project characteristics e.g. number of lines of source code, number of pages, number of links. Examples of algorithm models are COCOMO, SLIM and Function Points (FP) [3].

3) **Machine Learning (ML)** – Artificial Intelligence based techniques used as an alternative to EJ and AM more recently. Fuzzy logic, neural networks, regression trees and case-base reasoning (CBR) [3].

Expert Judgement relies heavily on past experience by software developers and project managers and is probably the first known way of estimating formally effort and cost estimation required to complete software projects.

The Construction Cost Model (COCOMO) is the first attempt at formulating an algorithmic approach to estimate effort and was developed and published by Barry Boehm in 1981, [15]) and was updated to COCOMO II in 1990, Boehm et al. [16]. Both techniques 'estimate effort by using statistical techniques based on exponential formulas adjusted using regression techniques' [17]. These proposed methods are all based on the notion of software size usually taken to mean Lines of Code (LOC) and is the most widely used method in the software industry, [18].

The Function Points measurement method was developed by Allan Albrecht at IBM and first published in 1979. This method was created as an alternative measure to Lines of Code and dealt more with functionality of a system as a means of determining size and hence effort and cost, [19].

The SLIM estimating model was developed by Larry Putnam of Quantitative Software Measurement in the late 1970s and used the Lines of Code (LOC) as a means of deriving size of a software project. These were modified using Rayleigh curve to determine estimates [20].

Case Based Reasoning (CBR) can be traced back to the early 1980s the work on Roger Schank in 1982 in a paper titled ‘Dynamic Memory: A theory of Reminding and Learning in Computers and People, [21]. CBR is an Artificial Intelligence based approach and the basic idea is that reasoning about new projects can be done from experiences or ideas from previous projects rather than starting to work from scratch, where ‘access to previous cases allows a problem solving and reasoning system to avoid known errors and to reach a solution in a guided way’. [21].

The techniques existed long before the World Wide Web existed and are being adopted and tailored to suit the needs of Web Engineering.

One such approach is the use of the Construction Cost Model (COCOMO) which is not specifically used for effort estimation for web applications since it is classified more as a generic algorithmic model. In the literature more regression-based algorithmic models are used. They are said to be ‘most suitable to local circumstances such as “in-house” analysis as they are derived from past data that often represents projects from the company itself’, [4]. The Web Engineering community is adopting a variant of the COCOMO models that uses regression techniques to derive estimation models.

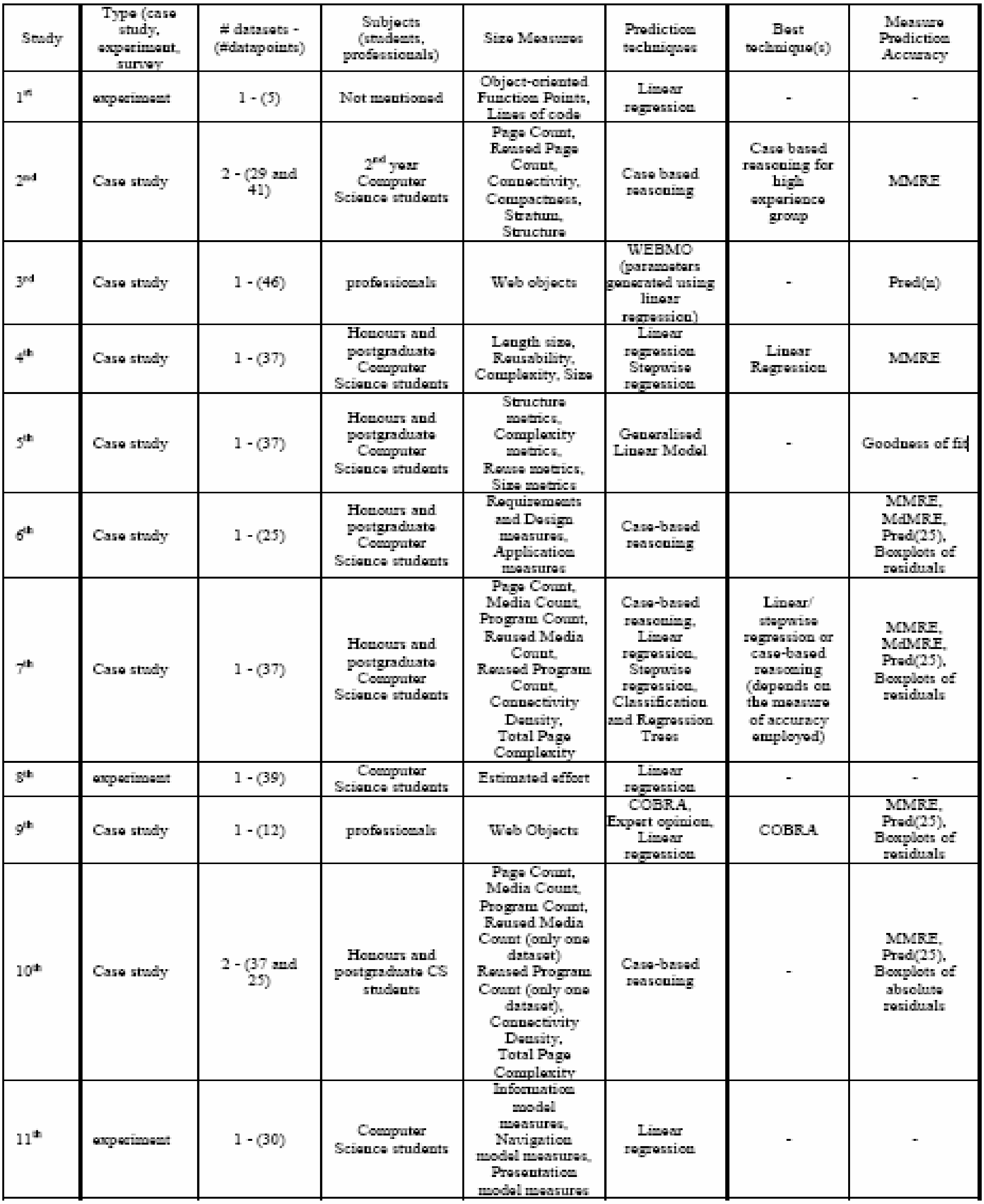
3. Application & Adoption of Traditional Software Effort Estimation Approaches to Web Engineering - Web Metrics, Data Sets & Modelling Techniques

A review of the literature (since 2000) - the time around which effort and cost estimation studies were first being explored, reveals that a wide range of web metrics, datasets upon which the metrics were extracted, and modelling techniques adopted from traditional software estimation, to derive ‘suitable’ models, were used.

A number of different size drivers intimate to web applications were the dominant metrics used to derive models e.g. number of web pages, number of images, total lines of codes and so on. Datasets were derived from a number of sources including student projects, within company past project data and cross company past project data as found in a number of databases. A number of techniques are found to have been used as well in different studies. In some cases the same techniques are being repeated in multiple studies whereas on the other hand, more than one technique was used in the same study. It was interesting to note that a fair amount of the data sets were gathered from projects conducted by students undertaking studies on a university course in web application development.

[22] surveyed the literature of studies completed between the 1995 and 2003 and presented a summary of the web metrics, datasets and modelling techniques used to develop effort and cost estimation models.

Table 1 below shows a summary of the studies surveyed by [22].



Study	Type (case study, experiment, survey	# datasets - (#datapoints)	Subjects (students, professionals)	Size Measures	Prediction techniques	Best technique(s)	Measure Prediction Accuracy
1st	experiment	1 - (5)	Not mentioned	Object-oriented Function Points, Lines of code	Linear regression	-	-
2nd	Case study	2 - (29 and 41)	2nd year Computer Science students	Page Count, Reused Page Count, Connectivity, Compactness, Stratum, Structure	Case based reasoning	Case based reasoning for high experience group	MMRE
3rd	Case study	1 - (46)	professionals	Web objects	WEBMO (parameters generated using linear regression)	-	Pred(n)
4th	Case study	1 - (37)	Honours and postgraduate Computer Science students	Length size, Reusability, Complexity, Size	Linear regression Stepwise regression	Linear Regression	MMRE
5th	Case study	1 - (37)	Honours and postgraduate Computer Science students	Structure metrics, Complexity metrics, Reuse metrics, Size metrics	Generalised Linear Model	-	Goodness of fit
6th	Case study	1 - (25)	Honours and postgraduate Computer Science students	Requirements and Design measures, Application measures	Case-based reasoning		MMRE, MdMRE, Pred(25), Boxplots of residuals
7th	Case study	1 - (37)	Honours and postgraduate Computer Science students	Page Count, Media Count, Program Count, Reused Media Count, Reused Program Count, Connectivity Density, Total Page Complexity	Case-based reasoning, Linear regression, Stepwise regression, Classification and Regression Trees	Linear/ stepwise regression or case-based reasoning (depends on the measure of accuracy employed)	MMRE, MdMRE, Pred(25), Boxplots of residuals
8th	experiment	1 - (39)	Computer Science students	Estimated effort	Linear regression	-	-
9th	Case study	1 - (12)	professionals	Web Objects	COBRA, Expert opinion, Linear regression	COBRA	MMRE, Pred(25), Boxplots of residuals
10th	Case study	2 - (37 and 25)	Honours and postgraduate CS students	Page Count, Media Count, Program Count, Reused Media Count (only one dataset) Reused Program Count (only one dataset), Connectivity Density, Total Page Complexity	Case-based reasoning	-	MMRE, Pred(25), Boxplots of absolute residuals
11th	experiment	1 - (30)	Computer Science students	Information model measures, Navigation model measures, Presentation model measures	Linear regression	-	-

Table 1 above showing summary of Effort and Cost Estimation Studies conducted by [22].

A careful analysis of table 1 shows a few trends as discussed by [22] such as:

- Size of the datasets used were generally small and ranged from 5 – 46 data points
- Size measures/metrics used for each study were not consistent through studies indicating a lack of standard in web metrics used.

- The types of web applications used in the empirical studies can be classified as web hypermedia applications and web software applications as described by [23], as cited in [22]. In other words, the web applications were not entirely similar in the sense that some were more static than others (dynamic) and as a result the technologies used to develop them were different. This may impact on the effort and cost required to develop the different types of applications for example, Web Hypermedia Applications are mainly developed using HTML, JavaScript and media whereas Web Software Applications would use dynamic components such as J2EE, PHP, and ASP.NET connected to a backend Database application typically.
- Most of the web applications understudy in this survey was developed by young and inexperienced programmers just out of university or in some cases while they were in university.

  Further observation reveals that:
- The dominant research study type was the case study. Very few experimental type studies were noted.
- The authors of these studies were involved in some way or the other in the development of the web applications in most cases.
- Function Points (FPs), Web Objects (WOS) and Lines of Code (LOC) were used as size drivers in a number of studies instead of the raw web metrics such as number of web pages etc. This is a more function-based approach.
- No mention is made of the use of any Software or Web Application Design and Development Process or Methodology, the size of the companies that developed these applications and the types of applications that they are developing.
- COCOMO or COCOMO II was not used in any of the studies.

Later studies conducted mainly used datasets from companies involved in web application development commercially. Less of students based approaches were used even though there is still some evidence of this method being used. More significantly however, is the use of databases that capture and store data about past web applications developed by various companies. These databases now allows for the development of effort and cost estimation models using cross company data instead of using data from within a single company alone.

In the literature, a number of these techniques and their variants, where possible, were deployed. [1] used COBRA (Cost Estimation, Benchmarking and Risk Analysis) – a tailored approach to COCOMO II used for Web Application Effort, in their work to derive a cost and effort estimation model. They suggest that this method be used when there is a lack of or small amounts of past project data. The further noted that COBRA itself ‘uses expert knowledge and a limited of past project data to predict a project’s development effort’. [1]. strongly believes that ‘standard software engineering approaches seem limited and cannot appropriately address all the challenges related to web development such as those listed in section 2.0. This search for more applicable approaches to effort estimation is evident in the cases discussed below.

[24] used Case-Based Reasoning (CBR), linear and stepwise regression techniques to estimate development effort for web applications. Mendes et al. (2002) applied a number of different configurations of CBR in their study of effort and cost estimation.

[18] developed a method called Chilean Web Application Development Effort Estimation (CWADEE) in order to get effort estimation in short periods such as within 24 – 72 hours. They claim that this method is ‘simple and specifically suited for small to medium-sized web based information systems’. CWAADEE introduced a new sizing metric called the Data Web Points (DWP) which they claim is ‘an indirect sizing metric [functionality] that takes into account the characteristics of Chilean experts, the available estimation time, the lack of great amounts of

historical information used to estimate'. This model is a highly theoretical one even though it is similar to Function Points and Web Objects. The authors have indicated that Computer Science graduate and undergraduate students in Software Engineering courses have been using CWADEE in the past three years for calculating the effort required for projects developed. The further suggested that even students with 'little experience can get good estimations using CWADEE'.

[25] conducted a study on the data produced by a small web application development company with 25 employees in Brazil. The aim of this study was to derive a 'simplified method to estimate the software size of web applications'. This company was using PHP, HTML, Java & MySQL to develop web applications. A simplified version of the IFPUG (International Function Point Users Group) Function Points was used as a means of deriving the size of web applications. Research data was taken from twenty web applications from the company.

[2] conducted a case study on 15 web applications developed by a medium-sized Italian Software Company with 15 employees. This company developed mainly enterprise information systems for government. Web projects used in this empirical study were classified as e-government, e-banking, web portals and intranet applications and were built using mainly J2EE, ASP, ASP.NET, Oracle etc. The measures considered for this study were those described by authors of previous studies e.g. number of web pages, new images, number of server side scripts. Additionally, the web objects approach to size measure was used. No mention was made of the experience of the employees of the company i.e. the skills or experience of the web application developers.

[26] conducted a replicated study [27] using a different experimental procedure that uses cross company data from the ISBSG database Release. They took a sample of 98 web software applications including 9 single-company projects and 89 cross-company projects. They applied Function Points, Language Type, Development Type and Platform as their main basis for size estimates. The main objective of this study was to predict whether cross-company models produce better estimates for effort and cost for within companies model. They found that 'predictions obtained for a single company using a cross-company model were significantly less accurate than those using its own (within-company) model.

[28] conducted another replicated study to investigate the accuracy of using cross-company estimation models to predict within-company models. The study used data from 83 web projects of the Tukutuku database. The Tukutuku project collects data about web projects globally and to use these datasets to build effort and cost estimation models and to benchmark productivity, [4]. Size measures used in these studies were obtained through the use of data from a survey using 133 online web forms aimed at giving quotes on Web Development Projects, [12]

These were varying motivations behind the studies briefly described. On one extreme, studies were conducted to derive newer, more suitable effort and cost estimation models, since it was believed that the models presented in the literature at the time were not suitable. On the other hand, replicated studies were conducted using datasets from the same databases as previous studies but with varying experimental procedures. Models were created using both cross and within company datasets. The cross-company models were used to compare how accurate they are at predicating efforts and cost for within-company models.

3.1. Effort & Cost Estimation/Prediction Techniques

The literature points to the use of a number of different techniques for deriving effort and cost estimation models for web applications i.e. a number of different techniques were used for predicting effort and cost estimation. Linear and Multiples Regression followed by Case Based

Reasoning are the two most popular techniques deployed. Other not so popularly deployed methods were Stepwise Regression, Expert Opinion, WebMO and COBRA, CWADEE.

[28] states that 'there is no clear answer to date as to what us the best technique to employ to obtain effort estimates, for a given dataset'. [29] as cited by [28] 'suggested that the dataset characteristics should have a strong influence on the choice of techniques to employ to obtain effort estimates'. They further suggested that 'the less "messy" the dataset i.e. small number of outliers, small amounts of collinearity, strong relationship between independent and dependent variables, the higher the chance that regression analysis will give the best estimation accuracy'. Conversely, 'very "messy" datasets should use case-based reasoning approaches to obtain more accurate effort estimates'.

[30] states that 'Linear Regression Analysis is one of the most commonly used statistical techniques for exploring the relationship between a dependent variable and one or more independent variables'. Table 1 above suggests that the most popular technique used for the studies surveyed between the periods 1999 - 2003 is the Linear Regression method. This probably indicates the belief that there was a strong linear relationship between the dependent and the independent variables under study.

Case-Based Reasoning (CBR) was also used as a prediction technique. [31] notes that 'the rationale for CBR is the use of historical information from completed projects with known effort. It involves 'characterising a new project p, which and estimate is required, with the attributes (features) common to those completed project stored in the case base [and the] use of characterisation as a basis for finding similar (analogous) completed projects, for which effort is known. This effort is achieved measuring the distance between two projects, based on the values of k features for these projects, [32] as cited in [31]. Unweighted Euclidean distance measures have been the most widely used in Software and Web Engineering[31].

One of the main issues with CBR is that it depends on the discovery of a project from the case base (repository) similar to the project being estimated. The case base is can be from the same company or from a cross-company database. There are also issues with algorithmic models such as Linear Regression Techniques. [12] suggest that 'effort estimations can be inaccurate whenever an algorithmic model, derived using past projects from one company, is used to generate estimations for projects belonging to a different company' [and that] 'inaccuracies occur whenever the relationship between effort and associated factors differs broadly from company to company'. They also suggested that CBR can be valuable where the domain is complex and difficult to model.

A number of later studies used other methods to predict effort and cost estimation for web applications. The studies that used these methods did not at all times clearly specify the reasoning behind the non-use of already existing methods.

[7] invented an approach that used Web Objects as a measure of size for effort estimation which he claims is a combination of expert judgement and data from 46 web projects using regression analysis. [25] used IFPUG Function Points as a means of deriving an estimated for effort and cost for their case study. [18] formulated an entirely new method called the Chilean Web Application Development Effort Estimation (CWADEE). Their motivation for deriving an entirely new method was based on their idea of having a model that can be used to derive estimated within time periods of 24 – 72 hours. Additionally, they have highlighted a number of issues that they believe were unique to their country Chile that might have made it unworthy of using existing methods to derive estimates. The invented a new size metric called Data Web Points as the basis for their new models.

3.2. Validation Methods & Predicting Accuracy of Models

The techniques described above allowed for the development of prediction models for effort and cost estimation. How are the prediction accuracy analysed for fitness of purpose and why is validating the prediction accuracy of these models and important issue? [33] suggests that if 'constructs cost estimation model using a particular data set, and then computes the accuracy of the model using the same data set, the accuracy evaluation will be optimistic'. To avoid this problem a number of cross-validation techniques were applied by different authors in the literature. Cross - validation is a common way of validating a cost estimation model. The basic idea is to use different subsets for model building (training sets) and model evaluation (test sets) [1].

In the studies surveyed, a number of variations of cross-validation were used. [2] adopted a leave-1-out cross-validation strategy by partitioning their dataset of 15 projects into two randomly selected sets of 14 projects for model building and the test set consisting of the remaining one project for model evaluation. [1] applied a similar strategy where they divided up their datasets of 12 projects into 11 projects for their training sets and the remaining 1 project for their evaluation set. [26] however, used a different combination of the cross-validation technique in their study which used 98 projects that included 9 single-company projects and 89 cross-company projects. They employed the following cross-validation combinations:

- 3- fold-cross validation [34] cited in [26] : single-company dataset was split into 3 different training and validation sets where the training sets had each 8 projects and validation sets had each 4 projects
- 6-fold-cross validation [35] cited in [26]: the single company data sets was split into 6 different training and validation sets, where the training sets had each 10 projects and the validation sets had each 2 projects
- Independent hold-out [36] cited in [26]: both cross and single-company models use the same validation set which is a subset of the single company dataset. Their hold out sample had 4 projects.

Their reasoning for using these various combinations was to investigate whether the predictions model generated from single-company data models were significantly better than those derived from cross-company data models.

To assess the acceptability and the prediction accuracy of the models a number of statistical techniques were deployed. The four most popular techniques as cited in [37] are:

- The Magnitude of Relative Error **(MRE)** [38]
- The Mean Magnitude of Relative Error **(MMRE)** [13]
- The Median Magnitude of Relative Error (**MdMRE**) [39]
- The Prediction at level n **(Pred(n))** [40]

In table 1 above the most popular methods deployed were the MMRE and Pred (25). Most of the studies presented in the literature use the MMRE, MdMRE and Pred (25) and these are often known to be the standard, de facto accuracy measures, [28]. However, [28] used box plots, Paired T-test and the Wilcoxon Signed Ranks test for two related samples in addition to 'de facto' standard accuracy measures listed. Box plots of residuals may provide a better insight on the effectiveness of a prediction model according to Kitchenham et al. (2006) as cited by [28]. This prompted the use of Box plots to measure the effectiveness of the prediction models.

All of the above listed approaches are well known and used in traditional software engineering estimation processes and were adopted for validation and prediction purposed in Web Applications Effort and Cost Estimation.

4. Discussion

A number of issues arise during the evaluation of the work completed so far on effort and cost estimation for web applications. These are mainly centred on the research type/method used, techniques to develop the models, the datasets employed and the notion of size and function of web applications and general validation issues.

Early studies focused attention on effort and cost estimation for web hypermedia applications instead of web software applications. This may have been influenced at the time by the relatively new nature of web software applications and technologies used to develop them in general. Datasets were generally taken from students projects at both at the undergraduate and post graduate level. [31] cited a number of studies that used data generated from students' development of web hypermedia applications. The general concern here is whether these datasets are valid enough to make generalizations beyond the domain for which they were developed? It should be noted too that these projects were done in the Education domain.

Other studies collected datasets from companies that were actually doing web application development on a commercial basis [1] [25][2] These studies used data from a single company to build effort and cost estimation models. The datasets were generally small – approximately 15 projects.

A more recent trend is the use of projects data from large database projects that are collecting data from web application development companies' world wide. The Tukutuku database project is one such example. The idea here is to have large enough datasets covering multiple companies across different countries so as to develop models that might have more general applications.

There is a great debate going on in the literature about the web metrics and size drivers that are most applicable for model building. The evidence of this is clear given no consensus has been made in the literature about which size drivers are more applicable for which types of applications. Some authors are more inclined to use static web metrics while others seem more inclined to use metrics that are based on functionality. [22] gives some insight into this. Whether it is the reluctance, lack of modelling and design process from which function based size metrics can be derived or just self belief that static measures are better approximation for size is not altogether clear. However, other authors seem more motivated to derive size metrics based on functionality.

[5] devised a metric called web objects which uses as its base, static measures such as number of web pages. [18] created what the called Data Web Points as their size driver for the method CWADEE. [25] seems more inclined to use variations of the proven Function Points technique as adopted from Software Engineering.

It is not always very clear in the literature as to why some researchers choose one method for deriving size over the other. However, at times there were indications that past experience and circumstances local to the researchers themselves may have influenced the size drivers used. In case of [22], it was reported that there was no other way of getting the data required for their analysis and possibly because they were developing web hypermedia applications which are less inclined to functionality measures than static size drivers.

A very important observation about the studies explored is that most of the studies were of the 'case study' type. Approximately 90% of the earlier studies revealed this fact as is noticeable in table 1. What are the implications for this and the results in general? In the literature, it is noted that 'a frequent criticism of case studies methodology is that its dependence on a single case renders it incapable of providing a generalizing conclusion' and that regardless if the sample size is 2, 10 or

100, it might just be impossible to make open generalizations - http://www.nova.edu/ssss/QR/QR3-2/tellis1.html

[22] points to this end by stating the following – 'although case studies are important and sometimes are the only way of obtaining not only data but also volunteers for the study, it is important to remember that their results only apply to the scope of the study [and that] the results cannot be generalized beyond the scope of the study'. The recommended the web engineering community to 'plan and run formal experiments' in order to build up a body of knowledge that can be generalized to a wider community.

The Literature indicates the use of a number of techniques for deriving effort and cost estimation/prediction models. Table 1 indicates a clear 'bias' towards the use of the Case Based Reasoning (CBR) approach. There are stated guidelines as to when a particular technique might be more applicable. CBR is popular for predicting estimations for a new project based on a case base populated with data from past projects. However, there are various ways of deriving variations of CBR as pointed out by [31]. Linear regression on the other hand is more applicable when there is thought of to be a kind of linear relationship between the dependent and the independent variables.

When these approaches are compared and contrasted to traditional software effort estimation a number of observations are spotted immediately. COCOMO II is quoted in the literature as being the most used method for estimating effort and cost, [42]. Lines of Code (LOC) and function points are still the dominant size drivers used even though in some case Use Case Points (UCP) is used. UCP is a variant of Function Points, [43].

The idea behind the scarce use of Lines of Code as a means of determining size of a Web Application is based on the fact that traditionally web applications were not traditionally written using much programming but mark-up languages such as HTML. However, this is no longer true as is noted from the definition of Web Software Applications. The move towards using functionality based size measures such as Web Objects and Data Web Points is an indication of this. Another reason for this could be due to the fact that web applications development rarely uses development methodology and as result design artefacts are not available as a means for estimating size.

5. Conclusion

This paper evaluated the body of research on effort and cost estimation models for web applications by examining the techniques that were used to build models, the datasets that were used and the research types employed. This was done in the context of adopting effort and cost estimation techniques from traditional software development.

Although many studies have been conducted on effort estimation models for web applications, there is no clear indication that there is a proven method or a set of proven methods for estimating the effort and cost of web applications. All of the techniques used are tailored versions of techniques taken from traditional software engineering. No significantly new techniques have been proposed.

Additionally, there is great debate about what size drivers should be used to derive estimates. More new size metrics are being developed and tailored from existing methods for e.g. Object Points, Web Objects, Data Web Points and more or less variations of the Function Points for which the reasons not always apparent. Other commonly used size drivers in traditional software development such as Lines of Code (LOC) are rarely used in Web Application Effort and Cost Estimation Models.

There are a number of outstanding issues that must be explored carefully before a company attempts to adapt this research. However, the body of work completed so far have explored many important issues and have provided many guidelines about the research results. This can be useful as guide and a base or starting point to companies considering the research results given that techniques are still being developed and are evolving to suit local needs. It is also most important to constantly re-examine effort and cost estimation approaches in modern day software development in the drive to develop tailored estimation techniques for web applications.

Future research should explore the possibility of using estimation approaches used in modern software engineering to estimate effort and cost for web applications due to a number of reasons:

- Web Applications are rapidly converging in their nature and characteristics to traditional software applications. More Web Applications are using heavy back end processing that requires programming intensive activities hence Lines of Code (LOC) should be considered as a metric
- There are more experienced web developers available today and working on Web Development Projects than a few years back and this may have an impact of effort and cost estimates
- There is evidence that Development Methodologies such as Agile Methods are being adapted by Web Application companies and this too may have an effect on estimate and cost
- Web Application Development Companies are following the trends of Global Software Development with dispersed teams
- Open Source Development Approaches and Tools are used more often.

References

[1] Ruhe, M, Jeffery, R., Wieczorek. I., 2003. Cost Estimation for Web Applications. IEEE Computer Society Press.

[2] Costagliola, G., Di Martino, S, Ferrucci, F, Gravino, C, Tortora, G, Vitiello, G., 2006. Effort Estimation Modeling Techniques: A Case Study for Web Applications. Procs of ICWE '06. ACM Press.

[3] Mendes, E., Mosley, N., Counsell, S., 2002. Comparison of length, complexity and functionality as size measures for predicting Web design and authoring effort. IEEE Procs on Software 149 (3), 86-92.

[4] Mendes, E., Mosley, N., Counsell, S., 2004. Investigating Web Size metrics for early Web Cost Estimation. ACM Press.

[5] Rollo, T., 2000. Sizing e-Commerce. Proc ACOSM 2000, Sydney, Australia.

[6] Baskerville, R.; Ramesh, B.; Levine, L.; Pries-Heje, J.; Slaughter, S. 2003. Is internet-speed software development different?, IEEE Software, 20(6), 70—77, November-December.

[7] Reifer, D., 2000. Web Development: Estimating Quick-to-Market Software. IEEE Computer Society.

[8] Standing, C. 2002. Methodologies for Developing Web Applications. Information and Software Technology, 44, 151--159.

[9] Pressman, R.S. What a tangled web we weave. IEEE Software, (Jan/Feb), 2000,18--21.

[10] Offutt, J. 2002. Quality attributes of Web software applications, IEEE Software, 19:2: 25-32, Mar/Apr 2002.

[11] Ambler, S.W., 2002. Lessons in Agility from Internet-based development. IEEE Software.

[12] Mendes, E., Mosley, N., Counsell. S., 2003. Investigating Early Web Size Measures for Web Cost Estimation. Procs of EASE 03 Conference, Keele. ACM Press.

[13] Shepperd, M.J., Schofield, C., and Kitchenham, B., 1996. Effort Estimation Using Analogy. Proc. ICSE-18, IEEE Computer Society Press, Berlin.

[14] Gray, A.R, MacDonell ,S.G, Shepperd,M.J: Factors Systematically Associated with Errors in Subjective Estimates of Software Development Effort: The Stability of Expert Judgment, Proceedings of the 6th International Symposium on Software Metrics, p.216, November 04-06, 1999

[15] Boehm, B.W., 1981. Software Engineering Economics. Prentice-Hall, Englewood Cliffs, N.J.

[16] Boehm, B.W., Horowitz, E, Madachy, R., Reifer, D., Clark, B.K., Steece, B., Brown, W.A., Chulani, S., Abts, C., 2000. Software Cost Estimation with COCOMO II. Prentice Hall PTR.

[17] Amor, J. J., Robles, G., Gonzalez-Barahona, J.M., 2006. Effort Estimation Developer Activity. Procs of EDSER 06, Shanghai, China. ACM Press.

[18] Ochoa, S.F., Bastaricca, M.C, Parra, G., 2003. Estimating the Development of Web Projects in Chile. IEEE Computer Society.

[19] Albrecht, A.J,, 1979. Measuring application development productivity. In Procs of the IBM Applications Development Symposium, GUIDE/ SHARE (Goiterey, Calic,‘Oct. 14-17). IBM.

[20] Putnam, L.H., 1978. General empirical solution to the macro software sizing and estimating problem. IEEE Trans. Soffw. Eng. SE 4, 4.

[21] Tsatsoulis, C., 1989. Case Based Design and Learning in Telecommunications. ACM Press.

[22] Mendes, E., and MOSLEY, N., 2005. Web Cost Estimation: principles and applications. Web Engineering – Principles and Techniques, Idea Group, Inc.

[23] Christodoulou,S.P; PA Zafiris,P.A, Papatheodorou T.S: 2000 - The Developers’ View and a Practitioner’s Approach to Web Engineering. Proceedings of Second ICSE Workshop on Web Engineering.

[24] Mendes, E., Counsell, S., Mosley, N., 2000. Measurement and effort prediction of web applications. Procs of the 2nd ICSE Workshop on Web Engineering, Limerick, Ireland. ACM Press.

[25] Candido, E.J.D, Sanches, R., 2004. Estimating the size of web applications by using a simplified function point methods. IEEE Software.

[26] Lokan, C., and Mendes, E., 2006. Cross-company and Single-company Effort Models Using the ISBSG Database: a Further Replicated Study. Procs ICESE 06, Rio de Janeiro, Brazil. ACM Press.

[27] Jeffery, R., Ruhe, M., and I. Wieczorek., 2001. Using public domain metrics to estimate software development effort. Procs Metrics'01, London.

[28] Mendes, E, Di Martino, S., Ferruci, F., Gravino, C., 2007. 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?. WWW 2007, Banff, Alberta, Canada. ACM Press.

[29] Shepperd,M.J;, G Kadoda, G. 2001 -_Using Simulation to Evaluate Prediction Techniques Proc. IEEE 7 International Software Metrics Symposium, *International Software Metrics Symposium*, London, UK,2001, pp. 349-358.

[30] Montgomery, D., Peck, E., Vining, G., 2001. Introduction to Linear Regression Analysis. John Wiley & Sons, Inc. 3. Ed.

[31] Mendes, E., Watson, I., Triggs, C., Mosley, N., Counsell, S., (2003a). A Comparative study of Cost Estimation Models for Web Hypermedia Applications. ACM Press.

[32] Angelis, L, Stamelos, I**:** A Simulation Tool for Efficient Analogy Based Cost Estimation - Empirical Software Engineering, **2000** - Springer

[33] Walkerden, F., Jeffery, R., 1999. An Empirical Study of Analogy based Software Effort Estimation. Empirical Software Engineering.

[34] Briand, L.C, Emam, K.E, Dagmar, S., Wieczorek, I,. Maxwell, K.D.: An assessment and comparison of common software cost estimation modeling techniques. Proceedings of the 21st international conference on Software engineering Los Angeles, California, United States , Pages: 313 – 322, 1999

[35] Briand, L.C, Langley, T. ,Wieczorek, I,. A replicated assessment and comparison of common software cost modeling techniques. Proceedings of the 22nd international conference on Software engineering, Limerick, Ireland, Pages: 377 – 386, 2000

[36] ***Lefley***, M. ***Shepperd***, M.J. "Using genetic programming to improve software effort estimation based on general data sets", Proc. GECCO ***2003***:

[37] Mendes, E., Mosley, N., Watson, I., 2002(a). A Comparison of Case-Based Reasoning Approaches to Web Hypermedia Project Cost Estimation. WWW2002, May 7-11 2002, Honolulu, Hawaii, USA. ACM Press.

[38] Kemerer, C. (1987), "An Empirical Validation of Software Cost Estimation Models," *Communications of the ACM*, May 1987, pp. 416-429.

[39] ***Myrtveit*** ,I, ***Stensrud***, E: A Controlled Experiment to Assess the ... the 6th International Symposium on Software Metrics, p.269, November 04-06, ***1999***

[40] M.J. Shepperd, and C. Schofield, "Estimating Software Project Effort Using Analogies." *IEEE Transactions on Software Engineering*, Vol. 23, No. 11, 1997, pp. 736-743.

[41] Kitchenham, B.A., Mendes, E., Travassos, G., 2006. A systematic review of cross- and within-company cost estimation studies. In: Proceedings of Empirical Assessment in Software Engineering. pp. 89-98.

[42] Keil, P., Paulish, D.J., Sangwan, R.S., 2006. Workshop papers: Cost estimation for global software development . ACM Press.

[43] Mohagheghi, P., Anda, B., Conradi, R., 2005. Empirical software engineering: Effort estimation of use cases for incremental large-scale software development. ACM Press.

Paper received: 2008-10-08