Effort Estimation in Early stages of Web Application Development

Manpreet kaur
Research Scholar, Department of Research Innovation & Consultancy
IKG PTU, Kapurthala, India

Dr. Sumesh Sood
Assistant Professor
IKG PTU Dinanagar Campus, India

Abstract: **Effort Estimation is process of estimating the development effort of web application. A cornerstone of Web project management is sound resource estimation. Resources are factors, such as cost, effort, quality, problem size, that have bearing on a project's outcome. Unfortunately, most Web development projects suffer from unrealistic project schedules, leading to applications that are rarely developed on time and within budget. Effort estimation consist in predict how many hours of work and how many workers are to develop a project. Estimation the project has not yet been solved and even the project manager has to deal with it since the beginning. Thus we need effective effort estimation techniques so that we can deal with these issues and develop web applications within budget and time meeting user requirements. Although estimating the effort required in developing web applications is a difficult task, accurate estimates of development effort have an important role to play in the successful management of web development projects. This paper is a review about web effort estimation methods and attributes from the other earlier researchers.**

Keywords: Web Applications, effort estimation, machine learning techniques.

I. INTRODUCTION

A. Introduction To Web Applications

Originally, Web was used by few scientists for sharing scientific information. But now, many of us rely on Web-based applications. A *web application* is defined as a software application that uses websites as a front end and back end provides full user functionality so that the user can affect the status of the business logic on the web server. The number of websites on internet is over one billion and is still increasing every year. But most of the web projects are not finished within time and budget by the Web developers. A survey on Web-based projects, published by the Cutter Consortium in 2000, revealed a number of problems with large outsourced Web-based projects (E. Mendes & Mosley, 2008):

- 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

B. Introduction to Web Effort Estimation

Effort estimation consists in predict how many hours of work and how many workers are needed to develop a project (M.J. Moayed, A. Ghani, 2007). Effort estimates help project managers allocate resources, control costs and schedule and improve current practices, leading to projects finished on time and within budget. Having realistic estimates of the required effort early in a Web application's life cycle lets project managers manage resources effectively (Emilia Mendes, Counsell, & Mosley, 2001). Numerous organizations world-wide are developing thousands of commercial and educational Web applications. However, there are no standardized development techniques or large datasets of historical data on Web development projects. For Web development, cost or effort is difficult to estimate because (Emilia Mendes, Mosley, & Counsell, 2005):

- There is no standard to sizing Web applications. Each can be created using diverse technologies such as several forms of Java, HTML, XML, and so on.
- People involved in Web development are represented by less experienced programmers (Reifer, 2002).
- Web project's primary goal is to bring quality applications to market as quickly as possible.
- Web development processes differ from traditional approaches (Reifer, 2002).

In the field of software development, effort estimation for conventional software projects, a number of methods have been developed, tested, and successfully implemented. But developing Web applications is different from conventional software projects. Three different methods: *Expert Judgement, Algorithmic Models and Machinelearning*, are used to estimate the effort required to complete web application projects. Most of the Web developers use previous similar project experiences or expert judgement for effort estimate. Examples of

algorithmic models are the COCOMO model, the SLIM model. Whereas, Machine Learning techniques are being used in aggregation or as replacements to algorithmic models. These methods include Neuro-fuzzy, Genetic Algorithms, Neural Networks, Fuzzy Logic and Regression trees, Case-based reasoning, Reasoning by Analogy etc. This becomes a problem to estimate how much effort needed to finish a web application project. False effort estimation can lead to a delayed project because there is not enough time to finish the project with the estimated effort. But there is "no silver" bullet method for effort estimation of web applications. Therefore, a method that can estimate effort needed by a web application project accurately has become important.

II. WEB EFFORT ESTIMATION PROCESS

A. Estimation Process Steps

Prediction or Estimation is a necessary part of an effective process, whether it be authoring, design, testing, or Web development as a whole. A prediction process involves(Mendes, E.; Mosley, N. & Counsell, 2001):

- The capturing of data about past projects or even past development phases within the same project.
- The identification of size metrics, cost drivers and the formulation of theories about their relationship with effort.
- The generation of prediction models to be applied to projects.
- The assessment of how effective those prediction models are.

B. Effort Estimation Techniques

The purpose of estimating effort is to predict the amount of effort to accomplish a given task, based on knowledge of other project characteristics that are believed to be related to effort. Project characteristics (i.e. independent variables) are the input, and effort (dependent variable) is the output we wish to predict as shown in figure 1.

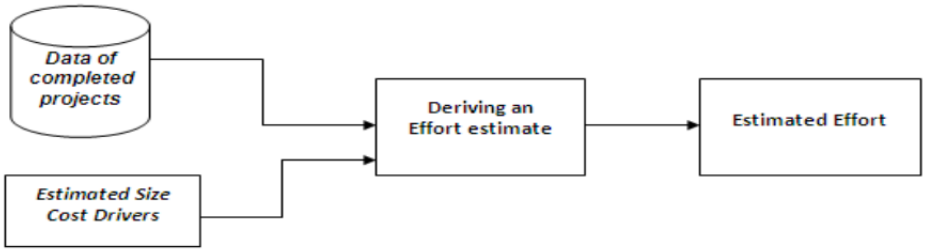

Figure 1: Components of Estimation Model

A task to be estimated can be as simple as developing a single function (e.g. creating a table on the database) or as complex as developing a large application, and in general the one input (independent variable) assumed to have the strongest influence on effort is size. Other independent variables may also be influential (e.g. developers' average experience, number of Web Effort Estimation tools employed) and these are often identified as cost drivers. Depending on the techniques employed, we can also use data on past finished projects to help estimate effort for new projects. Several techniques for effort estimation have been proposed over the past decades in software engineering and have been categorized into three categories as expert opinion, algorithmic models and artificial intelligence techniques.

i. Expert Opinion

Expert opinion represents the process of estimating effort by subjective means, and is often based on previous experience from developing/ managing similar projects. It has been and still is widely used in software and Web development. However, the means of deriving an estimate are not explicit and therefore not repeatable.

Limitations:

- Expert opinion, although always difficult to quantify, can be an effective estimating tool on its own or as an adjusting factor for algorithmic models.
- In Delphi method, no direct intervention is among the experts. Coordinator looks after the whole process.

ii. Algorithmic Techniques

The most popular techniques described in the effort estimation literature are algorithmic techniques. Such techniques attempt to formalise the relationship between effort and one or more project characteristics. The result is an algorithmic model. The central project characteristic used in such a model is usually taken to be some notion of software size (e.g. the number of lines of source code, number of Web pages, number of links). This formalisation is often translated as an equation:

$$Estimated\ Effort = a\ EstSizeNewproj^{b}\ EAF \quad \text{..... (1)}$$

where*a* and *b* are parameters that also need to be estimated. Equation (1) shows that size is the main factor contributing to effort, and can be adjusted according to an Effort Adjustment Factor (*EAF*), calculated from cost drivers (e.g. developers, experience, tools). An example of an algorithmic model that uses equation (1) is the Constructive COstMOdel (COCOMO) model.

Regression-based algorithmic models are 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. Regression analysis is used to generate regression-based algorithmic models, provides a procedure for determining the "best" straight-line fit to a set of project data that represents the relationship between effort (the response or dependent variable) and project characteristics (e.g. size, experience, tools, the predictor or independent variables).

Regarding the regression analysis itself, two of the most widely used techniques are multiple regression (MR) and stepwise regression (SWR). The difference between both is that MR obtains a regression line using all the independent variables at the same time, whereas SWR is a technique that examines different combinations of independent variables, looking for the best grouping to explain the greatest amount of variation in effort. Both use least squares regression, where the regression line selected is the one that reflects the minimum values of the sum of the squared errors. Errors are calculated as the difference between actual and estimated effort and are known as the residuals.

Limitations:

- The need for calibration of a model for each individual measurement environment.
- The variable accuracy level achieved even after calibration.

iii. Machine Learning Techniques

Machine Learning techniques have, in the last decade, been used as a complement to, or as an alternative to, the two categories discussed above. Some of the examples of machine learning techniques are fuzzy logic, regression trees, neural networks and case-based reasoning.

iv. Measuring Effort Prediction Power and Accuracy

An effort estimation model *m* uses historical data of finished projects to predict the effort of a new project. Some believe this is enough to provide accurate effort estimates. However, to gauge the accuracy of this model we need to measure its predictive accuracy.

To measure a model's predictive accuracy first calculate the predictive power for each of a set of new projects P_1 to P_n that used the effort estimation model *m*. Once predictive power for P_1 to P_n has been obtained, their values are aggregated, which gives the predictive power of model *m* and hence it's corresponding predictive accuracy.

v. Measuring Predictive Power

The most common approaches for measuring predictive power of effort estimation models are:

i. The Mean Magnitude of Relative Error (MMRE)
ii. The Median Magnitude of Relative Error (MdMRE)
iii. The Magnitude of Relative Error (MRE)
iv. The Prediction at level *n* (Pred(*n*))
v. Box plots of residuals

In order to validate a cost estimation model the predictive accuracy of a given effort estimation model needs to be calculated.

III. MOTIVATION

The estimated effort is a significant parameter used for the *cost estimation* of web applications. In the field of software development, a number of methods have been developed, tested, and successfully implemented for effort estimation of conventional software projects. Whereas, in case of Web application projects, there is need of improved effort estimation methods in order to complete them within budget and stipulated time.

IV. RELATED STUDY

The study of different research papers regarding effort estimation, comparison for web application and related domain from 2001 to 2017 is reviewed as under:

(Emilia Mendes et al., 2001)**,** It presents the use of analogy and two algorithmic models - linear regression and stepwise multiple regression to estimate the authoring effort of Web applications. Results suggest that estimation by analogy is a superior technique.

(E. Mendes and N. Mosley, 2001)**,** This paper presents two prediction models using statistical techniques, namely linear regression and stepwise multiple regression. The prediction power of the models was then compared. It suggests that stepwise regression gives better predictions than linear regression.

(E. Mendes, Watson, Triggs, Mosley, & Counsell, 2002)**,** The first is to compare the prediction accuracy of three CBR techniques to estimate the effort to develop Web hypermedia applications. The second objective therefore, is to compare the prediction accuracy of the best CBR technique, according to our findings, against three commonly used prediction models, namely multiple linear regression, stepwise regression and regression trees.

(Reifer, 2002), This paper proposed a new size metric i.e. *web objects* and implemented this metric using model WebMO, an algorithmic model,to estimate effort of web application.

(Watson, Mendes, Triggs, Mosley, & Counsell, 2002) This paper shows that "Weighted Euclidean Similarity Measure" gives the most accurate prediction accuracy among three different CBR techniques to predict the effort estimation of web hypermedia applications.

(Emilia Mendes & Counsell, 2002)This study compared the prediction accuracy of the best CBR technique, i.e. Weighted Euclidean Similarity Measure, against three prediction models, multiple linear regression, stepwise regression and regression trees. MMRE and MdMRE showed better prediction accuracy for Multiple Regression, whereas Boxplots showed better for CBR. This study showed that different measures of prediction accuracy give different results.

(E. Mendes, Mosley, & Counsell, 2002)**,** In this study, the researcher described the application of CBR for estimating Web hypermedia development effort using measures collected at different stages in the development cycle and compared their prediction accuracy. It shows that late measures did not show statistically significant better prediction than early measures.

(Emilia Mendes & Counsell, n.d., 2002)**,** This paper described the application of case-based reasoning to estimating the effort for developing Web hypermedia applications using early and late predictors and compared the prediction accuracy of different CBR configurations, using a Web hypermedia dataset.

(Ochoa, Bastarrica, & Parra, 2003) This paper proposed to fast estimate the development effort, using method CWADEE; in a short period of 24 to 72 hr using limited information. But its performance needs to be formally measured.

(Umbers et al., 2004), This paper proposed a simple, highly adaptable model using COSMIC FFP for application size measurement. This can be used by inexperienced estimation as well as average estimate. This hybrid model included both expert judgement and algorithmic techniques.

(E. Mendes & Kitchenham, 2004)**,** It investigated the use of analogy-based estimation with the use of two types of adaptation rules and contributed better estimation accuracy.

(Idril, Zakranil, Elkoutbil, &Abran, 2007) A modified COBRA i.e. Web-COBRA model was implemented on 12 web applications and Web-COBRA produced significantly better estimates than ordinary least squares regression and the subjective estimates. Experience shows that a cost estimation model that uses characteristics other than system size alone can also give a better effort estimate.

(Gu, Song, & Li, 2007)**,**This paper proposed few counting rules based on the characteristics of the system instead of the lines of code (LOC), and introduced an improved method i.e. COSMIC-FFP *like* for functional size measurement to estimate the effort of Web application. But it is not proved as a generalized model.

(M.J. Moayed, A. Ghani, 2007)The author investigated the relationship between size and effort of Web applications. It showed that Web Objects size measure gives results comparable to Function Points in case of large datasets.

(E. Mendes & Mosley, 2008), This paper showed that the use of simpler models, such as median effort, outperforms more complex models, such as Bayesian Networks. The only effective technique was MSWR (Manual Stepwise Regression).

(Ferrucci, Gravino, & Di Martino, 2008) In this paper, effort estimation was predicted using size measures *COSMIC* and *Web Objects*, using in combination with OLS Regression. The estimation results obtained were comparable.

(S. D. I. Martino, Ferrucci, Gravino, & Mendes, 2009), The author analyzed that size measure COSMIC-FFP is good for estimating Web application development effort, when used with Ordinary least Square regression modelling technique between two different datasets.

(Corazza, Di Martino, Ferrucci, Gravino, & Mendes, 2009)**,** The author investigated Web –COBRA profitably exploited for estimating Web application development effort, when used in combination with COSMIC.

(Lazić & Mastorakis, 2010), This paper proposed two improved models WEBMO+ and VPM+, which performed better than their original versions.

(Abrahão, De Marco, Ferrucci, Gravino, & Sarro, 2010)**,** A model-based size measure Object Oriented-Hypermedia Function Points (OO-HFP) was proposed and compared with standard function points analysis (FPA) method and thus gave better results than the later.

(Kaur, 2011)The author compared the Ordinary Least Square Regression and Case Based Reasoning techniques, using OO-HFP size measure and investigated case Based Reasoning outperformed OLS regression.

(Rosmina & Suharjito, 2012)**,** The researcher proposed a combined model FHSWebEE, consisting of OOmFPWeb, a functional size measurement by object-oriented based web application and Web metrics for web application size measurement. This showed better results than just using OOmFPWeb or Web metric.

(De Marco, Ferrucci, & Gravino, 2013)**,** The author proved that the use of COSMIC-based approximate counting is a suitable approach rather than the COSMIC average processes for early effort estimation.

(Ceke & Milasinovic, 2015)**,** This study made possible to predict web application development effort before application was even developed, not based on the application source code, but based on the not yet developed web application conceptual model. This study analyzes the possibility of using a combination of COSMIC, a functional size measurement method used to measure the size of software in terms of functional requirements requested by the user, and the conceptual model UWE. An effort model was built using simple linear regression analysis. Limitations: There is a need to large the employed dataset. There is also a need to develop an automated tool for CFP counting procedure used in UWE method.

(Čeke & Milašinović, 2015)**,**This paper proposed the frame of research on effort and cost estimation models for web applications. There is no strong suggestion that there is a certified method or a set of verified approaches for estimating the effort and cost of web applications. No ominously new techniques have been projected. New size metrics are tailored from current approaches Object Points, Web Objects, Data Web Points etc.

(Reza, Rahman, Parvez, Kaiser, & Al Mamun, 2015), In this study, the ISBSG dataset has been used for developing effort estimation models for web-based projects using IFPUG Function Point approach. Different machine learning techniques such as Stochastic Gradient Boosting and four SVR kernel techniques were employed. SVR techniques exhibit better results than other machine learning techniques.

(S. Di Martino, Ferrucci, Gravino, & Sarro, 2016)**,**In this paper, it is confirmed that the Web Objects method provides statistically superior results than the FPA method when used in combination with OLSR and Web – COBRA. But gives no statistically significant differences in the results obtained with OLSR, CBR, and Web-COBRA.

(S. Di Martino et al., 2016)**,** This study analyzes that the COSMIC method used to measure the "functional size" is more effective than Functional Point Analysis (FPA) for web effort estimation. But a smooth conversion from FPA to COSMIC is required.

(S. M. Satapathy, 2016)**,** Determining the functional size of software is one of the methods required to predict the size of the software to be developed, to plan the effort and resources required for development and to estimate a number of people to be employed for the task. This study proposed a tool named WADEES, for automated functional size estimation of web applications based on their conceptual models.

(Nagowah, Rumjaun, Gutteea, & Nagowah, 2017)**,** The paper presents some existing analogy-based software estimation tools used by project managers and these tools are critically analyzed to identify shortcomings. Finally an enhanced software effort estimation method is proposed. A system prototype named EffortEst has been implemented and evaluated based on the enhanced method. EffortEst provides the near-best estimation of software project effort with limited user intervention.

V. CONCLUSION

After analyzing the review in this paper, it is shown that there is an urgent need for adequate early stage effort prediction for web development For future directions, there is still a room for evolving new web estimation models which may lead to better effort estimates at early stages.We will study how early effort can be estimated for web applications using machine learning techniques as many such techniques like Support Vector Regression, Particle Swarm Optimization, Genetic Programming, Neural Networks, and Fuzzy Logic are under study. Because there is not any silver bullet or single method to early estimate the effort of web applications due to the varying technology, so machine learning techniques have a large scope to estimate effort successfully, as

these methods have been proved successful to solve optimization problems in the literature. The need of the hour is to have a model which can reduce the gap between approximate cost and actual cost of a web-application. Second, the effort/cost estimation model for web-application should help in approximating better estimates in the early stage of software development life cycle to allow project leaders or managers to monitor and manage the development process effectively and efficiently

REFERENCES

1. Abrahão, S., De Marco, L., Ferrucci, F., Gravino, C., & Sarro, F. (2010). A COSMIC measurement procedure for sizing web applications developed using the OO-H method. *Proceedings of the Workshop on Advances in Functional Size Measurement and Effort Estimation - FSM '10*, 1–8. https://doi.org/10.1145/1921705.1921707
2. Ceke, D. ., & Milasinovic, B. . (2015). Automated web application functional size estimation based on a conceptual model. *2015 23rd International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2015*, 234–241. https://doi.org/10.1109/SOFTCOM.2015.7314074
3. Čeke, D., & Milašinović, B. (2015). Early effort estimation in web application development. *Journal of Systems and Software*, *103*, 219–237. https://doi.org/10.1016/j.jss.2015.02.006
4. Corazza, A., Di Martino, S., Ferrucci, F., Gravino, C., & Mendes, E. (2009). Applying support vector regression for Web effort estimation using a cross-company dataset. In *2009 3rd International Symposium on Empirical Software Engineering and Measurement, ESEM 2009*. https://doi.org/10.1109/ESEM.2009.5315991
5. De Marco, L., Ferrucci, F., & Gravino, C. (2013). Approximate COSMIC size to early estimate web application development effort. *Proceedings - 39th Euromicro Conference Series on Software Engineering and Advanced Applications, SEAA 2013*, 349–356. https://doi.org/10.1109/SEAA.2013.41
6. Di Martino, S., Ferrucci, F., Gravino, C., & Sarro, F. (2016). Web Effort Estimation: Function Point Analysis vs. COSMIC. *Information and Software Technology*, *72*, 90–109. https://doi.org/10.1016/j.infsof.2015.12.001
7. E. Mendes and N. Mosley. (2001). Comparing Effort Prediction Models for Web Design and Authoring using Boxplots.
8. Ferrucci, F., Gravino, C., & Di Martino, S. (2008). A case study using web objects and COSMIC for effort estimation of web applications. In *EUROMICRO 2008 - Proceedings of the 34th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2008*. https://doi.org/10.1109/SEAA.2008.60
9. Gu, X., Song, G., & Li, Q. (2007). An improved FSM method for web-based applications. *CIMCA 2006: International Conference on Computational Intelligence for Modelling, Control and Automation, Jointly with IAWTIC 2006: International Conference on Intelligent Agents Web Technologies ...* https://doi.org/10.1109/CIMCA.2006.41
10. Idril, A., Zakranil, A., Elkoutbil, M., & Abran, A. (2007). Fuzzy Radial Basis Function Neural Networks for Web Applications Cost Estimation. *Proceedings of 4th International Conference onInnovations in Information Technology - IIT '07.*, 576–580.
11. Kaur, P. J. (2011). Comparative analysis of web effort estimation methods using OO-HFP as size measure, 59–68.
12. Lazić, L., & Mastorakis, N. E. (2010). Two novel effort estimation models based on quality metrics in Web projects. *WSEAS Transactions on Information Science and Applications*, *7*(7), 923–934. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-77955440090&partnerID=40&md5=1a8626944ee2c092bd8170563fe64c32
13. M.J. Moayed, A. Ghani, M. S. (2007). Comparing Between Web Application Effort Estimation Models. *Fifth International Conference on Computational Science and Applications*, 154–160. https://doi.org/10.1109/ICCSA.2007.33
14. Martino, S. D. I., Ferrucci, F., Gravino, C., & Mendes, E. (2009). Measures and techniques for effort estimation of web applications: An empirical study based on a single-company dataset. *Journal of Web Engineering*.
15. Mendes, E.; Mosley, N. & Counsell, S. (2001). Web Metrics — Estimating Design and Authoring Effort. *Web Engineering*, 50–57.
16. Mendes, E., & Counsell, S. (n.d.). A Comparison of Length , Complexity and Functionality as Size Measures for Predicting Web Design and Authoring Effort, 1–14.
17. Mendes, E., Counsell, S., & Mosley, N. (2001). Measurement and effort prediction for Web applications. *Web Engineering*, *2016*, 295–310. https://doi.org/10.1007/3-540-45144-7
18. Mendes, E., & Kitchenham, B. (2004). Further comparison of cross-company and within-company effort estimation models for Web applications. *10th International Symposium on Software Metrics, 2004. Proceedings.* https://doi.org/10.1109/METRIC.2004.1357920
19. Mendes, E., & Mosley, N. (2008). Bayesian Network Models for Web Effort Prediction: A Comparative Study. *IEEE Transactions on Software Engineering*, *34*(6), 723–737. https://doi.org/10.1109/TSE.2008.64
20. Mendes, E., Mosley, N., & Counsell, S. (2002). The application of case-based reasoning to early Web project cost estimation. *Proceedings 26th Annual International Computer Software and Applications*, 393–398. https://doi.org/10.1109/CMPSAC.2002.1045034
21. Mendes, E., Mosley, N., & Counsell, S. (2005). Investigating Web size metrics for early Web cost estimation. *Journal of Systems and Software*, *77*(2), 157–172. https://doi.org/10.1016/j.jss.2004.08.034
22. Mendes, E., Watson, I., Triggs, C., Mosley, N., & Counsell, S. (2002). A comparison of development effort estimation techniques for Web hypermedia applications. *Proceedings - International Software Metrics Symposium*, *2002–Janua*, 131–140. https://doi.org/10.1109/METRIC.2002.1011332
23. Nagowah, S. D., Rumjaun, S. S. B., Gutteea, K. A., & Nagowah, L. (2017). EffortEst- An Enhanced Software Effort Estimation by Analogy Method EffortEst- An Enhanced Software Effort Estimation by Analogy Method, (March).
24. Ochoa, S. F., Bastarrica, M. C., & Parra, G. (2003). Estimating the development effort of Web projects in Chile. *Proceedings - 1st Latin American Web Congress: Empowering Our Web, LA-WEB 2003*, (56 2), 114–122. https://doi.org/10.1109/LAWEB.2003.1250289
25. Reza, S. M., Rahman, M. M., Parvez, M. H., Kaiser, M. S., & Al Mamun, S. (2015). Innovative approach in web application effort & cost estimation using functional measurement type. *2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, (May), 1–7. https://doi.org/10.1109/ICEEICT.2015.7307462
26. Rosmina, & Suharjito. (2012). FHSWebEE: An effort estimation model for web application. *Procedia Engineering*, *50*(Icasce), 613–622. https://doi.org/10.1016/j.proeng.2012.10.067
27. S. M. Satapathy, S. K. R. (2016). Effort Estimation of Web-based Applications using Machine Learning Techniques. *International Conference on Advances in Computing, Communications and Informatics*, *Sept 21-24*.
28. Watson, I., Mendes, E., Triggs, C., Mosley, N., & Counsell, S. (2002). Using CBR to Estimate Development Effort for Web Hypermedia Applications, 1–5.