



Emerging Web Effort Estimation Methods: A Review

Manpreet Kaur

Deptt. of Computer Science, SRGCW,
Amritsar, Punjab, 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 application 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: *COCOMO, Function Points, Web Objects, Use-cases, Case- Based Reasoning(CBR).*

I. INTRODUCTION

Web Applications have evolved and became more complicated from time to time. Now a days, every web application can have different complexities. This becomes a problem to estimate how much effort is 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. The Web has been used as the delivery platform for two types of applications: Web hypermedia applications and Web software applications. A *Web hypermedia application*, is a non-conventional application characterized by the authoring of information using nodes, links, anchors, access structures and its delivery over the web. Technologies used for such applications are HTML, JavaScript and multimedia. It is developed to publish information over the web. A Web software application, is a conventional software application that depends on the web or use the web's infrastructure for execution. Such applications include databases, booking system, knowledge bases etc.

A variety of technological solutions are available for web developers to facilitate the delivery of quality web applications and bring them quickly to market. However, there are no standardized development techniques or large data sets of historical data on web development projects. For Web development, cost is difficult to estimate because:

- There is no standard to sizing Web application. Each can be created using diverse technologies such as several forms of Java (Java, Servlets, Applets and Java Server Pages), HTML, XML, and so on. Attempts have been made to apply "Function Points" principles to sizing web applications (Rollo, 2000; Mendes et al., 2002), however Mendes et al. did not find any improvement in estimation accuracy using Function Points based size metrics, when compared to other size metrics (Eg. Number of web pages, number of images).
- People involved in Web development are represented by less experienced programmers, users as developers, graphic designers and new hires straight from universities (Rollo, 2000; Standing 2002).
- Processes employed are in general ad-hoc (Standing 2002), although some organizations are starting to look into the use of agile methods (Amber, 2002).

Several techniques for cost and effort estimation have been proposed over the last 30 years in software engineering, falling into three general categories (Shepherd et al., 1996):

1) Expert judgment (EJ) – EJ has been widely used. 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 (Gray et al., 1999).
- In Delphi method, no direct intervention is among the experts. Coordinator looks after the whole process.

2) Algorithmic models (AM) – AM to date the most popular in the literature, attempt to represent the relationship between effort and one or more project characteristics. The main "cost driver" 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 pages, number of links). Algorithmic models need calibration or adjustment to local circumstances. Examples of algorithmic models are the COCOMO model (Boehm, 1981), the SLIM model (Putnam, 1978).

Limitations:

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

3) Machine learning (ML) - Machine learning techniques have in the last decade been used as a complement or alternative to the previous two categories. Examples include fuzzy logic models (Kumar et al., 1994), regression trees (Selby and Porter, 1998), neural networks (Srinivasan and Fisher, 1995), and case-based reasoning (Shepperd et al., 1996). A useful summary of these techniques is presented in (Gray and MacDonell, 1997b).

II. RELATED WORK

Last Web Estimation Methods

2.1 Function Point Method(FPM):The Function point method was developed by IBM. It is a combination of the analog method and weighing method. FPM is related with the function-oriented development methods and based on the user requirements concentrates on the functions. The effort estimation takes complexity, certain software characteristics and productivity into consideration. The Function Points is applied as a measuring means for the productivity used to specify the effort.

Strong Points of FPM:

- The function points as an estimation basis are detected by means of the type and complexity degree of the functions. The technical realization has no impact on the estimated value.
- FPM is a method to be applied at an early point to estimate efforts.
- The tool support for FPM may be helpful for small and average projects.

Weak Points of FPM:

- The FPM offers no breakdown of the specified effort to sub models, sub activities, sub products or functional units. Therefore, a micro estimation is only possible by means of empirical values, the utilization of an experience database, or the integration of other methods.
- Certain experience has to be expected for the application of the FPM with regard to function point determination.

2.2 Constructive Cost Model(CoCoMo):

It is a combination of parametric estimation equation and weighing method. Based on the estimated instructions, the effort is calculated by taking into consideration both the attempted quality and the productivity factors.

It is comprised of three levels:

2.2.1 Basic CoCoMo: By means of parametric estimation equations, the development effort and the development duration are calculated on the basis of the estimated DSI(Delivered Source Instructions).

2.2.2 Intermediate CoCoMo: The estimation equations are now taking into consideration 15 influence factors. These are product attributes, computer attributes, personnel attributes and project attributes. The degree of influence can be classified as very low, low, normal, high, very high, extra high .

2.2.3 Detailed CoCoMo: In this case, three levels of the product hierarchy(module, subsystem , system) and product – related influence factors are now taken into consideration.

Strong Points of CoCoMo:

- Macro estimation is done through Basic CoCoMo.
- Micro estimation is done through Intermediate and Detailed CoCoMo.
- Influential factors of CoCoMo model are characteristics of the project ,product, and the personnel as well the technology.

Weak Points of CoCoMo:

- Uncertainty of a DSI(Delivered Source Instructions) estimation.
- DSI are based on modern software engineering methods; no longer of great importance since the effort increasingly occurs during the early activities and DSI will only be effective towards the end of development process.
- The quantification of influence factors represent a certain problem , which has a strong impact on the quality of the estimation method and the required DSI information.
- Computer based support is required for Intermediate and Detailed CoCoMo, based on the quantity problem.

2.3 Web Objects:

Web Objects were proposed as a suitable measure of size for web applications. Reifer adds four new web-related components to the five function types of the Function Point approach : multimedia files, web building blocks, scripts, and links. The size of a web application is determined by evaluating the nine components of a web system based upon user requirements and page design(Moayed,Ghani,Mojtaba;2007).

Strong points of Web Objects:

- The metric used has a solid mathematical foundation.
- It can be easily extended to include new predictors as new elements are introduced for Web Applications(video markup languages, motion , sound etc.).
- The approach lets us address the unique characteristics of Web-based developments.

Weak points of Web Objects:

- Some in the metrics community would argue against the use of software science.
- The planning and data collection costs rise as you add predictors to handle new elements.
- Web object counts tend to be very sensitive to counting conversions.

III. NEW WEB ESTIMATION METHODS

3.1 Use-Cases (Robilio, Orosco;2008):

These are used to estimate effort in a simpler and more accurate way than the way in which it is done by using function points. Users are not completely satisfied with the results obtained by using Function points. One of the reason is more often than not, the time at which an estimated value can be obtained is too late to meet the investor’s needs. The two size measures used are: *Transactions* and *Entity Objects*.

Advantages:

- Use-cases can support an object-oriented design.
- They are widely used.
- They are the axis of the unified process, which employs a case-driven approach.

Table 1: Function Points vs. Use-cases

Function points	Use-Cases
To be consistent, the function points should be independent, but Kitchenham conclude that there are correlations among function point elements. Thus, elements can be counted more than once.	Independent
Another weakness of counting function points is its inconsistency. According to Kemerer, differences in function points of the same system averaged 12.2%,	More consistent
It computes the effort estimation at later stage.	It calculates the effort estimation at an early stage.
Difficult to count <i>Function Points</i> as they are correlated.	Easy to compute the two size measures: <i>Transactions</i> and <i>Entity Objects</i> .

3.2 Web Components or Web Objects (Das, Devdutta, Swain, Kumar;2011) :

Web Components are used to represent the size of Web application. Web components extend traditional Function Points by taking into account four additional types of objects : Multi-media files, Web building blocks, Scripts, and links. This model with extended function points proves to be more successful than the traditional function points. According to (Martino, Ferrunicci Gravino;2011), Ruhe et al. described two studies assessing the effectiveness of Web objects for estimating Web application development effort. Ordinary Least Squares Regression(OLSR) and Web COBRA. They also applied web objects as a size measure for Case based reasoning. It has been studied that Web objects performed well and better than the function points. It is also studied that Web objects gives better results when applied using Case-based reasoning than with OLSR.

3.3 Conceptual model based:

Web development is moving towards model-driven processes. This brings new opportunities to the Web project manager to make early estimates of the size and the effort required to produce web applications based on their conceptual models. This model uses a new size measure OO-HFP (Object-oriented hypermedia function point). According to (kaur, kritika;2011), this new size measure is applied on two estimation methods: OLSR(a algorithmic model) and Case based reasoning (a machine learning model). Here, it is observed that the new metric gives better results with Case based reasoning.

3.4 Case –Based Reasoning(CBR):

One of the machine learning approach is CBR, used for effort estimation of web applications. It is based on the psychological concepts of analogical reasoning, dynamic memory and the role of previous situations in learning and problem solving. It has advantages as follows:

- CBR has the potential to alleviate problems with calibration.
- CBR can be valuable where the domain is complex and difficult to model.
- The basis for an estimate can be easily understood.
- It can be used with partial knowledge of the target project.

- Intuitive and have a reasonable level of accuracy.
- CBR is also simple and flexible.
- May be applied to both qualitative and quantitative data, reflecting typical industrial datasets.

Disadvantages:

As with algorithm models, the effect of old data points is not clear. As an organization develops and successively introduces new technology, the older data points may become increasingly irrelevant and potentially misleading.

3.5 FHSWebEE:

According to (Rosmania, Suharjito;2012), a new size measure FHSWebEE is used, which is a combination of Functional size measurement i.e.OOmFPWeb (object-oriented function points for web applications) and web metrics .This new size measure is applied to estimation model case based reasoning. It also overcomes the limitation of using web metrics by Mendes. Web metric is not specific for object oriented based Web application. The researcher in (Rosmania, Suharjito;2012) shows that FHSWebEE is better than just using Function Points from OOmFPWeb and than web metrics from Mendes.

3.6 Combination of functional size and conceptual models:

The paper (Ceke, Milasinovic;2015) analyzes the possibility of using a combination of functional size and conceptual models for the purpose of web application development effort estimation. This approach to effort estimation was suitable and gave good effort estimation.

IV. DISCUSSION

- Whereas, Function Points might be inappropriate because applications do more than just transform inputs to outputs.
- Comparing “Function Points ” with “Web Objects” for measuring system size, it is revealed that the number of function points is lower than the number of Web Objects. It has been studied that the size difference between these two measures is up to 55% i.e. the number of web objects is twice as high as the number of Function Points. The sizing difference may have a large effect on the effort estimation accuracy.
- It is also studied that for larger projects, the difference between sizes measured with Web Objects versus Function Points increases.
- The results of all above discussed methods show that the new emerging techniques like case based reasoning gives better results than the traditional methods like Linear Regression.

V. CONCLUSION AND FUTURE SCOPE

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. Most of the effort estimation methods are based on traditional size measure Function Points. But the various emerging web effort estimation methods are using the extended versions of Function Points, which proves to be more successful. For future directions, there is still a room for evolving new web estimation models which may lead to better effort estimates at early stages. There is also a need to study various factors other than the web size measure which affect the cost and delivery schedule of web projects, for eg. , the effect of complexity, content organization, visual organization, navigation system, color, and typography.

REFERENCES

- [1] Robolio, Orosco, “Employing use cases to early estimate effort with simpler metrics”, Innovations system software Engineering, 2008.
- [2] Moayed, Ghani, Mojtaba, “Comparing between Web Application Effort Estimation Methods”, Fifth International Conference on Computational Science and applications, 2007, IEEE.
- [3] Das, Devdutta, Swain, Kumar, “Web Components as a measure for estimating Effort and Size of Web Applications”, IJCSIT, Vol.2 (3), 2011, 1137-1143.
- [4] Kaur, Kritika, “Comparative Analysis of web effort estimation methods using OO_HFP as size measure”
- [5] Rosmania, Suharjito, “FHSWebEE: An Effort Estimation Model for Web application”, ICASCE 2012, Elsevier, 613-622.
- [6] Ceke, Milasinovic, “Early effort estimation in Web application development”, Elsevier, 2015, 219-237.
- [7] Mendes, Mosley, “Web Metrics- Estimating Design and Authoring Effort”, Web Engineering(IEEE), 2001.
- [8] Mendes, Mosley, “Investigating Web Size Metrics for Web Cost Estimation”, The journal of systems and software(Science Direct), 2004.
- [9] Umbers, Miles, “Resource Estimation for Web Applications”, IEEE, 2004.
- [10] Vigo, Brajnik, “Automatic Web Accessibility Metrics: where we are and where can go”, Elsevier, 2011.
- [11] Mendes, Counsell, “Web Development effort Estimation using Analogy”, IEEE, 2000.
- [12] Letchmunam, Roper, wood, “Investigating effort prediction of web based application using CBR ”, 2004.