The Journal of Systems and Software 103 (2015) 219–237

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Early effort estimation in web application development

Denis Čeke[a,*], Boris Milašinović[b]

[a] Quality Assurance Office, University of Tuzla, Bosnia and Herzegovina
[b] Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia

ARTICLE INFO

Article history:
Received 3 July 2014
Revised 31 December 2014
Accepted 6 February 2015
Available online 11 February 2015

Keywords:
Web Application
Function points
Effort estimation

ABSTRACT

Project planning in software industry represents one of the most complex tasks, especially when there is a need to estimate the time, cost and effort needed for development of software projects. In the field of development effort estimation for classical software projects a number of methods have been developed, tested and successfully implemented. Web projects are, by their nature, different than classical software projects, and there is a lack of methods and models that provides a high degree of confidence in development effort estimation. This paper analyzes the possibility of using a combination of functional size and conceptual models for the purpose of web application development effort estimation. Measurement of functional size can be effectively applied to the conceptual models of the data-driven web applications because of the existence of extensive count of data movements. For the purpose of this study 19 web applications with their conceptual models were employed. An effort model was built using simple linear regression analysis. Upon construction, evaluation and validation of the effort model prediction accuracy elements, R^2, MMRE, and Pred(l), showed promising results for web projects used in the model construction and validation process.

1. Introduction

Almost every organization today has a corporate site and/or commerce site in order to provide information to the public and/or to support their business needs. Due to the current trends in the business world, there is an increasing demand for more complex web applications. However, web application managers and developers face problems when they need to estimate development time, effort and cost of the projects on the basis of customer's requirements. In the field of Web Engineering, effort estimation represents one of the crucial problems as there is no "*silver bullet*" method for effort estimation. Software size measures in traditional software engineering are well investigated and there are quite a number of methodologies and practices on how to estimate the size and effort in classical software application development (Attarzadeh and Hock Ow, 2010; Basha and Ponnurangam, 2010; Muhairat et al., 2010; Zia et al., 2012). However, this is not the case with web applications because their development is different from traditional software development (Abrahão et al., 2010; Baresi and Morasca, 2007; Reifer, 2000; Baskerville et al., 2003). Traditional software size and effort estimation techniques are not adequate to capture specific features of the development that can influence the size and effort required for the development of web applications (Costagliola et al., 2006).

The primary goal of web application conceptual model is to help in the process of designing and implementing web application and to serve as a guideline for verification of the application requirements. As stated in Ginige (2008), the process of web application development can be viewed as transformation of conceptual model into a physical system. Web application models need to be descriptive, as much as possible, in order to be able to capture all characteristics of the elements that need to be implemented in the web application. This requires tools that have the ability to create models with sufficient expressiveness in the web application domain, such as the MagicUWE tool ("MagicUWE").

There are generally two approaches for sizing web applications (Candido and Sanches, 2004): lines of code (LOC) and functional size (FS). There are also some other custom solutions that have been illustrated in Azhar et al. (2012). There is an ongoing debate in the literature about the use of LOC for sizing web applications because LOC depends on the programming language and it is applicable only once the project is finished. On the other hand, functional size estimation can be applied in the early stage of development.

Functional size measurement (FSM) is a concept on how to measure the size of software in terms of functional requirements requested by a user. The first method that was developed to support this concept was function point analysis (FPA) invented by Albrecht (1979). In his study, Albrecht defined a function point (FP) as a unit of measure that represents the amount of business functionality that

* Corresponding author at: dr.Tihomila Markovića br 1, 75000 Tuzla, Bosnia and Herzegovina. Tel.: +387 35 278 350. Fax: +38735278350

E-mail addresses: denis.ceke@untz.ba, denis.ceke@fer.hr, ceke.denis@gmail.com (D. Čeke), boris.milasinovic@fer.hr (B. Milašinović).

http://dx.doi.org/10.1016/j.jss.2015.02.006
0164-1212/

an information system provides to a user. Since then, many standards have been developed following his concept. Four main standards that have developed into ISO standards are: IFPUG ("ISO, ISO/IEC 20926: Software engineering – IFPUG 4.2 unadjusted functional size measurement method – counting practices manual," 2004), Mark II ("ISO, ISO/IEC 20968: Software engineering – Mk II function point analysis – counting practices manual," 2002), NESMA ("ISO, ISO/IEC 24570: Software engineering – NESMA functional size measurement method version 2.1 – definitions and counting guidelines for the application of function points analysis," 2005) and COSMIC ("The COSMIC Functional Size Measurement Method Version 3.0.1 Measurement Manual," 2009).

FSM techniques can be used as software sizing methods in effort estimation. The advantage of these methods lies in the fact that they are independent of technology or programming language used and can be used through the entire development life cycle (Trendowicz and Jeffery, 2014). With FSM methods we can estimate the size of a software application and, thus, the development effort of the software application at the beginning of the development process, which might not be the case with other methods.

When we combine conceptual models and FSM methods, the former for modeling user requirements and the latter for sizing software, we get a powerful tool for effort estimation in the earliest stages of software development, that is, requirements analysis.

In De Marco et al. (2013) the authors have analyzed the studies using 1st generation of FSM methods for software sizing and highlighted some difficulties in their application suggesting the use of other FSM methods for sizing web applications, like the COSMIC method. Also, in De Marco et al. (2013) the authors have examined the studies that were encouraging the use of the COSMIC method for web application effort estimation. The COSMIC method has strong applicability in estimating the size of the multi-layer software architecture. It is easy to learn, stable and cost-effective in implementation. The method is well accepted by the staff involved in the estimation process because of its ease for creating mapping rules to modern software requirements documentation methods and its compatibility with the modern software architectures. With this method, it is possible to make improvements in estimation accuracy, especially for large projects. Also, it has the ability to size software based on the requirements created in CASE tool ("Advantages of the COSMIC Method," 2014, "The COSMIC Functional Size Measurement Method Version 4.0, Introduction to the COSMIC Method of Measuring Software," 2014). In the past few years, researchers have suggested several techniques and measures to be used in the application models for the purpose of effort estimation of the software, such as COCOMO, SLIM, SEER etc. (Boehm et al., 2000; Catal and Aktas, 2011).

Researchers were trying to adopt FP to be used in web applications (Costagliola et al., 2006). The COSMIC-FFP (Common Software Measurement International Consortium-Full Function Points), now known as COSMIC, is a software sizing method for measuring a "functional size" of the software for the case of sizing real-time and business applications ("The COSMIC Functional Size Measurement Method Version 3.0.1 Measurement Manual," 2009). The method is based on counting data movements that are mostly represented in business software and the biggest programming effort is usually spent on development and implementation of such features (Costagliola et al., 2006; "The COSMIC Functional Size Measurement Method Version 3.0.1 Measurement Manual," 2009).

Web application development based on the conceptual models is becoming very popular because these models can be reviewed and corrected before they are turned in to "live" applications. Managers and developers can benefit from this approach because it allows detailed analysis of an application before writing the program code.

For modeling web applications, researchers have suggested several model-driven approaches, such as OOHDM (Schwabe and Rossi, 1995), WebML (Ceri et al., 2000), OO-H (Gómez et al., 2001), W2000 (Baresi et al., 2000), and UWE (Knapp et al., 2004). However, in the context of effort estimation in relation with a model-driven development, the main problem of how to estimate effort based on a conceptual model remains unresolved (Abrahão et al., 2010).

Most of the approaches that have been suggested for sizing web applications on the basis of functional points have some limitations; for instance, they were relaying on implementation decisions (Abrahão et al., 2010), they were not automated (Candido and Sanches, 2004; Cleary, 2000; Costagliola et al., 2006), they were not developed in accordance with any standards or did not take into account some new features that resulted from technological developments. There have been several proposals on how to apply the COSMIC method to estimate a functional size of object-oriented applications (Diab et al., 2001, 2002; Poels, 2003) and web applications (Costagliola et al., 2006; Mendes et al., 2002; Rollo, 2000; Umbers and Miles, 2004). Our approach is slightly different than others, and it consists of model-driven approach for modelling application with the UWE (Knapp et al., 2004) methodology and the COSMIC function points method for counting data movements on the basis of the UWE model which is described in Section 5.

The use of model-driven development ensures all the benefits of the methodology while the UWE approach utilizes separation of concerns (content, navigation, process etc.). This allows us to divide the main web application model into sub models, from requirements analysis to process structure. We used four models in the UWE approach: use case, content, navigation and process. Every model in the UWE approach has its mission in completing the tasks concerning a transformation from user requirements to final implementation model. We applied COSMIC FP counting procedure on the UWE business process model in the web application conceptual schema because this model is derived from all previous models and it can be assumed, after the analysis of all previous models, that the UWE business process model is the most appropriate for counting data movements. In order to evaluate an effectiveness of the proposed method, we conducted empirical validation on 10 web applications developed by the professional developers from a software development company and 9 web applications developed by the professional developers employed at the university IT department.

In Section 6, we employed simple linear regression analysis for the creation of the effort estimation equation where we estimated the invested effort for developing web applications. In generated equation, CFP was used as an independent variable (predictor) for the purpose of building an effort estimation model.

2. The COSMIC method

The COSMIC measurement method is based on the principle that the functional user requirements are formed from a collection of functional processes. Every functional process is triggered either by a user or initiated by an actor (a functional user or an external component) that occurs outside the boundary of the software that is measured. This request initiates data group movement which can consist of one or more attributes that belong to that group. The process is complete when the software has done all that is needed in order to respond to the event ("The COSMIC Functional Size Measurement Method Version 3.0.1 Measurement Manual," 2009).

COSMIC method recognizes four types of data movements: Entry, Exit, Read and Write.

- *Entry* – moves data from functional users to functional process.
- *Exit* – moves data from functional process, across boundary to functional users.
- *Read* – moves data from persistent storage to functional process.
- *Write* – moves data from functional process to persistent storage.

Each of these data movements is viewed as one COSMIC function point (CFP). After all functional processes have been finished,