

Effort Estimation for the Development of Web Portals

Sergio di Martino

Università di Salerno, Italy

Filomena Ferrucci

Università di Salerno, Italy

Carmine Gravino

Università di Salerno, Italy

INTRODUCTION

In recent years, the availability of server-side, Web-oriented component technologies, such as Enterprise Java Beans, ASP.NET, SOAP, and so forth, has led to profound changes in the scenario of information systems, allowing developers to create enterprise Web applications, that is, highly-dynamic information systems able to deliver a complex amount of functionalities, while running in a Web browser. Among those Web applications, the *Web portals* are of special interest. Basically, a portal is a Web site collecting related information and/or services from different sources. It is intended to provide features such as personalization, integration of applications and business intelligence, notification (push technology), and infrastructure functionality (Winkler, 2005). Due to their intrinsic advantages, Web portals are becoming an essential support for the activities of lots of organizations, companies and public institutions.

From a developer point-of-view, it is recognized that portal development is very similar to normal Web application development, since they mainly differ in the contents. However, as complexity and size of Web applications/portals increased, their development introduced a set of unique features and characteristics, making it far different from traditional software programming (Ginige & Murugesan, 2001). Among the main issues, we can find: the requirements are instable; portals are usually characterized by pressure time and compressed development schedule; the employed technologies rapidly change (technology instability); and portals are usually developed by a small team including young developers, with different backgrounds and knowledge compared to a traditional software development team. The discipline of Web Engineering was aimed at addressing these specific issues, supporting complex Web Application design, development, evolution, and evaluation (Ginige & Murugesan, 2001). However a lot of research is still in progress. In particular, currently many works are addressing the crucial problem of early estimating the effort required to develop enterprise Web applications. Indeed, development

effort (the work carried out by software engineers and developers) is the dominant project cost, being also the most difficult to estimate and control, with significant effects on the overall costs. Thus, effort estimation is a critical activity for planning and monitoring software project development and for delivering the product on time and within budget. Significant over or underestimates can be very expensive and deleterious for a company. Thus, it is paramount for the competitiveness of a Web company to be able to effectively predict in advance the effort required to develop an enterprise Web-based project (Costagliola et al., 2005; Mendes, Mosley, & Counsell, 2002).

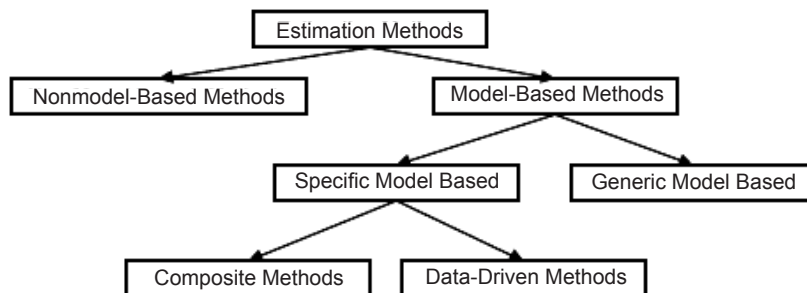
In the article, we will introduce the main concepts and problems behind the issue of effort estimation for the development of Web portals. Moreover, we will provide an insight on the current state of the art about existing proposals, in particular those that can be seen as an extension of function points, and finally we will outline the major future trends in this field.

BACKGROUND

In the literature a lot of different methods to estimate software development cost have been proposed. In Figure 1, it is depicted a widely accepted taxonomy of estimation methods. While *nonmodel-based methods* mainly take into account expert judgments (thus with highly subjective factors), *model-based* approaches involve the application of some algorithms to a number of inputs to produce an effort estimation.

The inputs for these kinds of algorithms are some factors able to heavily influence the resulting development effort of a software project. Among these, *software size* is accepted as a key cost driver, since it has deep influences of total development efforts, and thus on total project costs (Boehm et al., 2000). Consequently, being able to obtain an early size measure for a project can provide a significant estimation of the overall development costs. In the context of traditional

Figure 1. Classification of estimation methods (Briand & Wieczorek, 2002)



desktop applications, several size measures have been conceived to be employed in effort/cost models to predict the cost needed to design and implement the software, and then to manage their development. Among them, *function points (FPs)* (Albrecht, 1979) have achieved a wide acceptance to estimate the size of business systems.

In recent years, many researchers and practitioners tried to apply these approaches to Web applications, but soon it was recognized that these methods were not adequate, being both unable to capture the specific features affecting the size and the effort required for Web systems (Morisio, Stamelos, Spahos, & Romano, 1999; Rollo, 2000; Rhue, Jeffery, & Wieczorek, 2003b), and difficult to apply (Rollo, 2000). However these approaches have many inherent appealing features, since they allow the prediction of the effort, cost, and duration of a project, based only on the *functional user requirements (FURs)*. This has motivated recent proposals of adaptation/extension of these methods to the Web domain.

In the following we will provide an insight on the main Web-oriented methods.

EFFORT ESTIMATION

In recent years, some size measures have been specifically conceived, especially to size Web applications. Among them, of special interest, are *Web objects* (Reifer, 2000; Ruhe, Jeffery, & Wieczorek, 2003a), and *COSMIC-FPP* (Costagliola et al., 2005; Mendes et al., 2002), which represent adaptations or extensions of software measures defined in the context of traditional desktop applications. Some were encouraging, yet initial empirical validations of these measures have been provided (Costagliola et al., 2005; Ruhe et al., 2003a, 2003b).

Web Objects

Web objects represent an extension of *FPs*, which are briefly recalling in the following. The current standard definition

and counting procedure of the *FP* approach is reported in the *IFPUG Counting Practices Manual* (IFPUG, 2001). The measurement of the system size starts with the identification of all the functions, which can be of the following types: *external input*, *external output*, *external inquiry*, *internal logical file* and *external interface file*. The first three classes are considered transaction function types while the last two are considered data function types. Then, these identified functions are weighted in agreement with standard values specified in the *Counting Practices Manual*, by using their types and the level of their complexity.

The *Web objects* approach extends *FPs* by introducing four new Web-related components (*multimedia files*, *Web building blocks*, *scripts* and *links*), used as predictors together with the five traditional function types of *FPs* (Reifer, 2001). The number of *Web objects* is determined by evaluating the nine components in functional user requirements. Following the estimation procedure, first the instances of the components are counted and their complexity (low, average, high) is determined. Then, by using a calculation worksheet, a weight is associated to each counted instance. Thus, the functional size of a Web application in terms of *Web objects* is given by the sum of these weights (Reifer, 2001).

Ruhe et al. (2003a) provided an empirical study on the application of *Web objects*, and results showed better performance for the proposed measure with respect to standard *Function Points*. Subsequently, two cost estimation models based on *Web objects* have been proposed: the *WebMO* (Reifer, 2000) and the *WebCOBRA* (Ruhe et al., 2003b).

WebMO

WebMO cost model is a direct extension of the *COCOMO II* early design model. Let us recall that *COCOMO II* (CONstructive COSt Model) is a parametric model used to estimate effort and schedule for software development projects (Boehm et al., 2000). *WebMO* exploits its underlying ideas in the context of Web applications. It does not require a deepened knowledge about the influence of cost drivers

4 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/effort-estimation-development-web-portals/17886

Related Content

A Multi-Objective Genetic Algorithm for Software Personnel Staffing for HCIM Solutions

Enrique Jiménez-Domingo, Ricardo Colomo-Palacios and Juan Miguel Gómez-Berbís (2014). *International Journal of Web Portals* (pp. 26-41).

www.irma-international.org/article/a-multi-objective-genetic-algorithm-for-software-personnel-staffing-for-hcim-solutions/123172

A Formal Approach for the Validation of Web Service Orchestrations

Wael Sellami, Hatem Hadj Kacem and Ahmed Hadj Kacem (2013). *International Journal of Web Portals* (pp. 41-54).

www.irma-international.org/article/formal-approach-validation-web-service/78352

Enhancing the Portal Experience

Joe Lamantia (2010). *International Journal of Web Portals* (pp. 12-25).

www.irma-international.org/article/enhancing-portal-experience/44693

Digital Rights Protection Management of Web Portals Content

Theodoros Evdoridis (2007). *Encyclopedia of Portal Technologies and Applications* (pp. 256-263).

www.irma-international.org/chapter/digital-rights-protection-management-web/17879

Challenges of Multi Device Support with Portals

Jaye Fitzgerald and Van Landrum (2012). *Enhancing Enterprise and Service-Oriented Architectures with Advanced Web Portal Technologies* (pp. 87-102).

www.irma-international.org/chapter/challenges-multi-device-support-portals/63947