



Cost estimation in software engineering projects with web components development

Javier de Andrés ^a, Daniel Fernández-Lanvin ^b & Pedro Lorca ^c

^a Accounting Department, University of Oviedo, Asturias, España jdandres@uniovi.es

^b Department of Computer Sciences, University of Oviedo, Asturias, España dflanvin@uniovi.es

^c Accounting Department, University of Oviedo, Asturias, España plorca@uniovi.es

Received: May 7th, 2014. Received in revised form: April 16th, 2015. Accepted: May 4th, 2015.

Abstract

A variety of models for the prediction of effort costs in software projects have been developed, including some that are specific for Web applications. In this research we tried to determine if the use of specific models is justified by comparing the cost behavior of Web and non-Web projects. We focused on two aspects of the cost calculation: the diseconomies of scale in software development and the impact of some project features that are used as cost drivers. We hypothesized that for these kinds of projects, diseconomies of scale are higher but the cost-increasing effect of the cost drivers is mitigated. We tested such hypotheses using a set of real projects. Our results suggest that both hypotheses hold. Thus, the present research's main contribution to the literature is that the development of specific models for the estimation of effort costs for the case of Web developments is justified.

Keywords: software projects; Web; costs; estimation.

Estimación de costes en proyectos de ingeniería software con desarrollo de componentes web

Resumen

Existen multitud de modelos propuestos para la predicción de costes en proyectos de software, algunos orientados específicamente para proyectos Web. Este trabajo analiza si los modelos específicos para proyectos Web están justificados, examinando el comportamiento diferencial de los costes entre proyectos de desarrollo software Web y no Web. Se analizan dos aspectos del cálculo de costes: las diseconomías de escala, y el impacto de algunas características de estos proyectos que son utilizadas como cost drivers. Se enuncian dos hipótesis: (a) en estos proyectos las diseconomías de escala son mayores y (b) el incremento de coste que provocan los cost drivers es menor para los proyectos Web. Se contrastaron estas hipótesis analizando un conjunto de proyectos reales. Los resultados sugieren que ambas hipótesis se cumplen. Por lo tanto, la principal contribución a la literatura de esta investigación es que el desarrollo de modelos específicos para los proyectos Web está justificado.

Palabras clave: proyectos software; Web; costes; estimación.

1. Introduction

The research question we try to address in the present paper is whether Web projects behave in a different way than non-Web software projects with respect to certain issues that have an impact on effort costs. Effort costs are the costs for paying software engineers, and they are harder to assess in the earlier stages of a project [1]. An interesting avenue of research is devoted to the development of models for the estimation of

effort costs. Some models use expert judgments. Some others rely on the use of heuristic procedures based on statistical/machine learning methods. Lastly, others imply theoretical assumptions, for example those that stem from econometric theory. Despite these different approaches, most of the models do not take into consideration the specific features of the different types of projects.

In this regard, the development of a Web application has specific features caused by both technical and social/cultural

reasons. So, several specific measures have been proposed by different authors to design accurate effort estimations, specifically in Web projects. These authors propose and compare estimation techniques for Web applications development using industrial and student-based datasets [2–11]. All these methods are based on the assumption that the effort costs of Web development projects follow patterns that are certainly different from those of traditional software developments [12] and, hence, require differently tailored measures for accurate effort estimation [6,13–16]. We try to shed some light on whether such assumption is true.

This question is important because if the behavior of Web projects is not significantly different from that of non-Web projects, then the development of methods that are Web-specific is unadvisable because these models are developed using only a subset of the available data. This will lead to less accurate models.

Accuracy of cost estimations when managing software projects is extremely important because good effort estimates lead to projects finished on time and within budget [17]. An estimate of the costs is desired even during the earlier stages of the development [1]. Hu et al. in [18] indicate that a 200 to 300 percent cost overrun and a 100 percent schedule slippage would not be unusual in large software systems development projects. Millions of dollars have been wasted in projects that are abandoned because of severe cost overruns and schedule slippages [19].

Furthermore, the study of the specific case of Web projects is important because, during the last several years, Web developments have been representing an increasing share of software projects. This has been caused by the diffusion of the Internet, and has generated a market and therefore a demand for Web applications.

We undertook a comparative analysis on a massive database which is made up of both Web and non-Web projects. Specifically, we assessed the effect of the Web nature of a project on a) economies/diseconomies of scale, and b) the effect on effort costs of multipliers usually considered in cost models such as those capturing the product, hardware, personnel and project attributes. We conducted a series of regression analysis by means of which we tried to assess if the coefficients defining the impact of such factors on costs are significantly different for Web and no-Web projects.

Our results indicate that a) the diseconomies of scale are significantly higher for the case of Web developments, and b) the specific features of Web developments have an influence on the aggregate effect of the cost drivers, which cause the cost of projects to be significantly lower. Thus, the main contribution of the present research to the literature is that we provide evidence that the development of specific models for the case of Web projects is clearly justified. This opens further avenues of research, such as the study of the effect of the Web nature of projects on each one of the cost drivers defining the features of a software project.

The remainder of this paper is structured as follows: Section 2 discusses why Web projects are different; Section 3 analyzes the effort implications of the specific features of Web projects; Section 4 is devoted to the design of the analysis; and Section 5 outlines the main results. Finally, a summary, conclusions and future avenues of research are detailed in section 6.

2. Web development is different

Practical experience in web development shows that there are significant differences between traditional software applications and web applications [9]. There are many factors that differentiate a web development from common software development projects. The most evident are the strong requirements that are imposed by the technical environment. The use of HTTP protocol, the complexity of web user interfaces with regard to the user experience, or the heterogeneous pool of components, frameworks and protocols present in a common web application [16,20] are just some of the factors that determine the complexity of the development. Furthermore, social and cultural factors must also be taken into account while estimating these kind of projects: the usual skills of web developers, the kind of projects and their features and many others that can also have influence in the final cost of the project.

Regarding the project management, Web developments are characterized by having a very fluidic scope [21,22]. The development teams are small (from three to seven developers) [13,20,22,23] and they usually work intensively [24] and under a high pressure [14]. Their members are mostly inexperienced novice young programmers [16,20,23]. Another problematic aspect is related to the software requirements, given that in Web projects are very volatile [14,23] and that the development of web applications is characterized by the fact that their requirements generally cannot be estimated beforehand, which means that project size and cost cannot be anticipated either.

In relation to the development processes, these are usually ad-hoc or heuristic, although some organizations are starting to look into the use of agile methods [25] like Scrum or Extreme Programming (XP). This agile approach suits the unstable requirement environment that was previously pointed out [26], given that agile methods are based on the assumption that software processes should be based of adaptation to changes in the requirements, as these changes will happen from time to time [27].

From a more technical point of view, the diversity of technologies in these developments is high compared with the rest of software projects. They are usually highly component oriented (mashups, frameworks, application servers, etc.), and can be created using diverse technologies such as several varieties of Java (Java, servlets, Enterprise java Beans, applets, and Java Server Pages), HTML, JavaScript, XML, XSL, etc. [16,20]. These two aspects can have positive and negative influences on the cost of the project. Integrating third-party tested components and elements can save a lot of resources since we can expect them to work adequately, waiving the cost of not only the building, but also the testing of these parts of the product. However, the integration can be complicated, and finding developers with the required skills is usually difficult.

3. Different features of the techniques proposed for cost and effort estimation.

The features previously presented differentiate Web developments from the rest of software projects. But, how

do they determine the cost of the project? It will now be analyzed what can be expected if any of the popular software estimation cost strategies were applied. The different techniques proposed for cost and effort estimation over the last 30 years fall into three general categories [16]: (i) expert judgment, (ii) machine learning and (iii) algorithmic theory-based models (AM). The latter is, to date, the most popular in the literature, and they 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, functional points) [28]. Algorithmic models need calibration or adjustment to local circumstances.

The most popular representative of the AM is the Constructive Cost Model (COCOMO), developed by Boehm et al. [29]. COCOMO has three increasing detailed and accurate forms. The basic form estimates effort using the following exponential function:

$$Y = a(KLOC)^q \tag{1}$$

Where Y is the effort in person-months, $KLOC$ is the software size measured in terms of thousands of lines of code, and a and q are constants determined by the environment and the complexity of the application to be developed. Intermediate and detailed versions of COCOMO incorporate an adjustment multiplicative factor that depends on a subjective assessment of product, hardware, personnel and project attributes, which are understood as cost drivers.

It is interesting to highlight that COCOMO establishes an exponential relationship between software size and effort. Such an exponential relationship allows for the modeling of economies and diseconomies of scale. However, in COCOMO q is always bigger than 1, so it always hypothesizes diseconomies of scale in the software development process.

The exponential approach has later been used in other models such as COCOMO II [30] (and its multiple adaptations like the proposal of Patil et al. [31]) and Constructive Systems Engineering Cost Model (COSYSMO) [32]. The main difference between COCOMO, COCOMO II and COSYSMO is the way they calculate the size of the project and the factors that we must consider to calibrate the adjustment.

In this research we will consider COSYSMO to explain the behavior that Web development projects would play when we apply an algorithmic model to estimate the costs. This is because in COSYSMO the assessment of the cost drivers resembles the current circumstances in software development in a more accurate way. COSYSMO uses the following exponential function:

$$PM_{NS} = A \times Size^E \times \prod_{i=1}^n EM_i \tag{2}$$

Where PM_{NS} is the effort in person-months, A is a calibration constant derived from historical project data, E represents diseconomies of scale and EM is an effort

multiplier for each cost driver (the geometric product results in an overall effort adjustment factor to the nominal effort).

Table 1.
Multipliers in COSYSMO models.

1	Requirements Understanding	The level of understanding of the system requirements by all stakeholders (A stakeholder in COSYSMO is any primary or secondary actor that could have any interest or collaboration in the development of the project) including the systems, software, hardware, customers, team members, users, etc....
2	Architecture Complexity	The relative difficulty of determining and managing the system architecture in terms of platforms, standards, components (COTS/GOTS/NDI/new), connectors (protocols), and constraints. This includes systems analysis, tradeoff analysis, modeling, simulation, case studies, etc....
3	Level of Service Requirements	This cost driver rates the difficulty and criticality of satisfying the ensemble of level of service requirements, such as security, safety, response time, interoperability, maintainability, Key Performance Parameters (KPPs), the “ilities” (The “ilities” may include: reliability, usability, performance, Affordability, maintainability, and so forth. The “ilities” are imperatives of the external world as expressed at the boundaries with the internal world of the system [35].), etc.
4	Migration Complexity	The complexity of migrating the system from previous system components, databases, workflows, etc., due to new technology introductions, planned upgrades, increased performance, business process reengineering etc....
5	Technology maturity/ Technology risks	The relative readiness of the key technologies for operational use.
6	Documentation to Match Lifecycle Needs	The formality and detail of documentation required to be formally delivered based on the life cycle needs of the system.
7	# and Diversity of installations/platforms	The number of different platforms that the system will be hosted and installed on.
8	# of Recursive levels in the design	Number of applicable levels of the Work Breakdown Structure
9	Stakeholder Team Cohesion	Leadership, frequency of meetings, shared vision, approval cycles, group dynamics (self-directed teams, project engineers/managers), IPT framework, and effective team dynamics.
10	Personnel Capability	Systems Engineering’s ability to perform in their duties and the quality of human capital.
11	Personnel Experience/Continuity	The applicability and consistency of the staff over the life of the project with respect to the customer, user, technology, domain, etc....
12	Process Maturity	Maturity per EIA/IS 731, SE CMM or CMMI.
13	Multisite Coordination	Location of stakeholders, team members, resources (travel).
14	Tool Support	Use of tools in the System Engineering environment.

Source: Adapted from Valerdi [36,37].

Table 2.
Rating Scales for COSYSMO Effort Multipliers.

	Very Low	Low	Nominal	High	Very High	Extra High	EMR
Requirement Understanding	1.87	1.37	1.00	0.77	0.60	--	3.12
Architecture Understanding	1.64	1.28	1.00	0.81	0.65	--	2.52
Level of Service Requirements	0.62	0.79	1.00	1.36	1.85	--	2.98
Migration Complexity	--	--	1.00	1.25	1.55	1.93	1.93
Technology Risk	0.67	0.82	1.00	1.32	1.75	--	2.61
Documentation	0.78	0.88	1.00	1.13	1.28	--	1.64
# and diversity of installations/platforms	--	--	1.00	1.23	1.52	1.87	1.87
# of recursive levels in the design	0.76	0.87	1.00	1.21	1.47	--	1.93
Stakeholder team cohesion	1.50	1.22	1.00	0.81	0.65	--	2.31
Personnel/team capability	1.50	1.22	1.00	0.81	0.65	--	2.31
Personnel experience/continuity	1.48	1.22	1.00	0.82	0.67	--	2.21
Process capability	1.47	1.21	1.00	0.88	0.77	0.68	2.16
Multisite coordination	1.39	1.18	1.00	0.90	0.80	0.72	1.93
Tool support	1.39	1.18	1.00	0.85	0.72	--	1.93

Source: Adapted from Boehm et al. [29]

The parameter E determines the behavior of the cost in relation to the size of the project. Typical determining factors are learning effect (the bigger the project is, the longer the time that developers have to adapt to the technology, environment, etc.) and coordination costs. Regarding this, we could expect an important weight of the learning effect.

Working with novice developers, immature and heterogeneous technologies lead us to expect an important period of developer adaptation to the project in which productivity would not be as high. This adaptation period does not depend of the functional size of the project, so its pernicious influence would be smaller the bigger the project is, and this would be reflected with an economic effect in the estimation. Furthermore, the fact that Web developments are usually driven by ad-hoc and agile methods with small teams will probably have the opposite effect. Even their promoters state that agile methods perfectly suit small projects with small teams [33]. Part of their agility is based on the simplification of the coordination tasks, something possible when we have small teams, but that can be risky when these are bigger. For instance, Scrum suggests groups no bigger than nine people [34], while XP suggests that the smaller is the team, the better [27]. Consequently, we could expect increasing coordination problems as long as the project size grows. This may cause diseconomies of scale to be more relevant for the case of Web projects.

In relation with the multipliers, Table 1 summarizes the factors considered in COSYSMO, and Table 2 shows the rating scales to be applied once the subjective assessment of the factors has been made. Some of these factors have a direct relationship with the effort required, while for others the relationship is inverse. The bigger the requirements and

architecture understanding, the stakeholder and team cohesion, team's capability and experience, process capability, multisite coordination or tool support are, the smaller the cost of the project will be. However, level of service requirements, migration complexity, technology diversity and risks, documentation and recursive levels in the design exert a direct influence in the cost of the project. It will now be considered how the facts that were previously discussed regarding Web projects can be determined. At first sight, these features will probably lead to a higher cost of Web developments. The volatility of requirements involves a low level of requirement understanding (1), and will probably also determine the architecture understanding (2). The high technological complexity and heterogeneity of Web projects is directly represented in the diversity of installation and platforms factor (7), increases the migration complexity factor (4) and the technology risk/maturity (5). The latter is also determined by the experience of the developers. Novice developers will not be able to anticipate technological problems (like for example, integration) derived from the combination of different tools and standards. This fact also has an influence on the team capability and experience factors (9 and 10).

Furthermore, some features of Web development such as the more frequent use of agile methods or the intensive component oriented development level may mitigate the impact of some of the factors on the total cost. For instance, the pernicious effect of a low understanding of the requirements is not a problem in Scrum or XP, since both are designed for this kind of scenarios. The way these methods organize the project encourage the team to have straight communication between developers and stakeholders (factor 8), as well as their physical organization (factor 12). For example, XP and Scrum state that customers and developers should work together and meet continuously throughout the project. Another determining use of agile methods is the simplicity of the deliverables, reducing the documentation to the minimum (factor 6). Finally, the generalized use of frameworks, third party components and specially the deployment over application servers that provide an important part of the logic already implemented suppose a high tool support (13) that should reduce the cost of the project.

In summary, the specific features of Web developments have both positive and negative effects on the cost estimation. However, given the growing adoption of agile methods have in the industry [38], and the mitigation effect they have in the penalizing effects, we think that there are reasons to suppose that the cost-reducing features of Web projects outweigh the cost-increasing ones.

4. Hypotheses

According to what was discussed in Section 0, we can formulate two different hypotheses. On one hand, the literature points out that Web projects are mainly undertaken by small teams that do not require advanced coordination mechanisms, which makes them vulnerable to any increment in the size of the project. This leads us to formulate the following hypothesis:

H1: The diseconomies of scale are significantly higher for the case of Web developments.

Moreover, even when some of the specific features of Web development would involve a negative effect in the cost, others features like the adoption of agile development methods in these projects can have a mitigation effect over the former. Thus, a second hypothesis can be formulated:

H2: The specific features of Web developments have an influence on the aggregate effect of the cost drivers that cause the cost of projects to be significantly lower.

5. Methodology

In this section we explain the empirical model we used to test our hypothesis and we provide a description of the database we used.

5.1. The model

As we have previously noted the COSYSMO model proposes an exponential equation for the computation of the effort. This equation can be summarized as:

$$Effort = a(Size)^b m(X) \quad (3)$$

Where b represents the diseconomies of scale and $m(X)$ is the aggregate effect of the multiplier factors which act as cost drivers. In order to test H1 and H2, which state that for the case of Web applications both the scale effects and the aggregate effect of the multipliers are significantly different, we formulate a modified version of (3) which takes the following form:

$$Effort = a(Size)^{(b+c*web)} [(1 + d * web)m(X)] \quad (4)$$

Where web is a dummy that equals 1 for web projects and 0 otherwise. If c and d are significantly different from zero and, respectively, positive and negative then H1 and H2 hold. If these parameters are not significantly different from zero then the Web nature of a project should have no influence on the cost estimation process.

Equation 4 can be transformed into a linear equation by considering it in logarithmic form:

$$\ln(Effort) = \ln(a) + (b + c * web)\ln(Size) + \ln[(1 + d * web)m(X)] \quad (5)$$

Rearranging the terms in (5) we obtain:

$$\ln(Effort) = \ln(a) + \ln[m(X)] + b\ln(Size) + cweb\ln(Size) + \ln(1 + d * web) \quad (6)$$

If $d=0$, then $\ln(1+d*web)=0$. Otherwise, the value of $\ln(1+d*web)$ depends on web , that is, on whether or not the considered project is a Web application. In order to reach an

expression which can be estimated through linear regression we replace $\ln(1+d*web)$ by $Kweb$, where K is a continuous variable to be estimated. If K is significantly different from zero then d is also significantly different from zero, so K can be used to test H2.

As is detailed below, the database used for empirical testing does not have information on the specific assessment of the individual cost drivers for each project. So, we consider the effect of such cost drivers on a fixed form. In other words, in our model they form a constant. Then, we can add it to the prior constant in the model and define a new intercept term in the equation which takes the form $Intercept = \ln(a) + \ln[m(X)]$. So, the final equation to be estimated is:

$$\ln(Effort) = Intercept + b\ln(Size) + cweb\ln(Size) + Kweb \quad (7)$$

In equation 7 the parameters to be estimated are: Intercept, b , c , K . As indicated above, c and K are the parameters of interest for the H1 and H2 tests, respectively.

5.2. The database

For the test of our model we used the ISBSG (International Software Benchmarking Standards Group) Development & Enhancement Repository, dated June 2009. This is a huge database that comprises of 5052 projects from twenty-four countries. The ISBSG Repository is a consolidated data set in the field of software metrics (Further details can be found at <http://www.isbsg.org/>). It has been used in many papers which deal with the issue of effort estimation [39–42]. It has also been used for the study of other topics related to software metrics, such as software defect prediction, [43] the determination of the optimum development team size [44] and the determination of the priorities of the Capability Maturity Model Integration (CMMI) process areas [45].

It is noticeable that the ISBSG Repository does not contain information on the values of the factors used as cost drivers in COCOMO, COSYSMO and others. As we indicated above, in our equation such factors are aggregated and included into the intercept term. So, in our study we are limited to assess the impact of the Web nature of a project on the global effect of all the multipliers. However, there are two reasons that prevented us from considering databases containing the assessment of cost drivers for each of the individual projects:

a) Such databases do not contain information on whether the software projects are Web applications or not. In most cases they are old databases containing projects which were developed in years prior to the popularization of the Internet.

b) The size of most databases is too small and does not allow an accurate estimation of the parameters of individual cost drivers. Is interesting to remember that COSYSMO uses 14 cost drivers, and if we want to capture the differential effect of the Web nature of the project we need 14 additional coefficients. Modeling the intercept term and the diseconomies of scale requires another three parameters. So, 31 variables are needed for the development of a detailed model. This, under the usual standards, requires a sample size of 300 or more for an accurate estimation of the parameter

(some databases used for the development of effort estimation models can be found on the PROMISE Software Engineering Repository), which is publicly available at [46]. Examples of databases in this repository which have one or more of the mentioned features that make them unsuitable for our study are COCOMO 81, COCOMO NASA, COCOMO NASA 2 and Desharnais Software cost estimation dataset).

An additional advantage of the ISBSG Repository is that it contains information on the quality of the data gathered on each project. A rating code from A to D is assigned to each project depending on the quality of the data. The submission is fundamentally sound only for the projects that are A or B-rated. So, in order to ensure high reliability in our results we discard projects with a C or D rating code. This procedure is in accordance with prior studies that used the ISBSG Repository (see, e.g., Pendharkar et al.,[47], among others).

Another issue which may bias the research results is the measurement of software size. The ISBSG repository provides information of the functional sizing of each project, as other measures such as the lines of code can be influenced by the language used and the programmer’s characteristics. However, there are different approaches for the measurement of the functional size: that proposed by the International Function Point Users Group (IFPUG), the functional sizing of the Common Software Measurement International Consortium (COSMIC), that of the Netherlands Software Metrics Association (NESMA) and the MK II method by the UK Software Metrics Association, among others.

Many prior papers on effort estimation pool all the projects in the same database no matter the method used for effort estimation. This is a source of bias, since different methods generate not equivalent estimations, even restricting the study to those covered in the ISO/IEC 14143-1:2007 (IFPUG, COSMIC, NESMA and MKII). In fact, particular attention over the past several years has been devoted to finding a mathematical function for converting IFPUG functional size units to the newer COSMIC ones [48]. So, we decided to restrict the study to those projects for which IFPUG functional sizing –the most popular one- was available.

Table 3.

Variables used in the model

Variable	Definition
EFFORT	Number of hours needed for the completion of the project.
Web	Web nature of a project. 1 for Web applications and 0 otherwise.
ADJFP	Size of the Project, measured in IFPUG functional points.

Source: Elaborated by the authors.

Furthermore, the temporal scope considered for our study was the period 1999-2009. The reason for this selection is to make the comparison between Web and non-Web projects on a homogeneous basis, as Web technologies are of recent development. Web developments previous to 1999 are mainly implemented using now deprecated technologies like CGI or the older versions of PHP. The roughness of these technologies strongly determined the development projects. In the same way, considering non-Web projects from years prior to the popularization of the Web could cause significant biases in our research. After this last filter, the final sample consists of 588 projects.

With regard to the rest of the variables (apart from software sizing) used to test our model, first we defined which projects are considered as Web developments. We only considered as Web projects those that were developed in a language using a virtual machine (Java, Java/similar, J2EE, C#, .Net.). We discarded other kinds of projects such as web page design or similar because they cannot be considered as Web application developments and therefore, they are beyond the scope of this research. As indicated in the prior subsection, the Web nature of the projects was represented through a dummy variable that equals 1 for Web applications and 0 otherwise.

With respect to the effort, it was measured through the Summary Work Effort field of the ISBSG repository. This variable provides the total effort in hours recorded against the project. Table 3 provides a summary of the variables used in our model and their codification hereafter.

Table 4.

Sample breakdown by year and type of project.

Year	New developments			Enhancement			Total		
	Web=0	Web=1	Total	Web=0	Web=1	Total	Web=0	Web=1	Total
1999	1	0	1	0	0	0	1	0	1
2000	2	2	4	3	4	7	5	6	11
2001	2	7	9	3	0	3	5	7	12
2002	1	12	13	6	9	15	7	21	28
2003	4	2	6	3	5	8	7	7	14
2004	6	38	44	17	71	88	23	109	132
2005	9	11	20	3	9	12	12	20	32
2006	3	99	102	2	236	238	5	335	340
2007	3	3	6	5	3	8	8	6	14
2008	2	0	2	1	0	1	3	0	3
2009	0	0	0	1	0	1	1	0	1
TOTAL	33	174	207	44	337	381	77	511	588
t-test									
Mean	2004.21	2004.94	2004.82	2003.89	2005.34	2005.17	2004.03	2005.20	2005.04
Std Dev	2.247	1.570	1.710	2.148	1.197	1.415	2.182	1.348	1.533
t-stat		-1.773			-4.395			-4.597	
p-value		0.084			0.000			0.000	

Source: Elaborated by the authors.

6. Results

As the database and the variables have been defined, we must indicate that we carried out three separate analyses: one for the total sample, another one considering only enhancement projects and another in which only projects which consisted of the development of new software were included. The reason for this is that enhancement projects can involve different circumstances for example lack of documentation, refactoring requirements and legacy systems compatibility restrictions, that could strongly determine the cost effort required by the project beyond the web or non-web condition.

Table 4 shows the sample breakdown by year and type of project. We also included the results of a t-test of the difference of means, for Web and non-Web projects, for the year of the completion of the project.

It is noticeable that the majority of the projects are Web developments, both in the new developments and in the enhancement subsamples. Most of the projects are from the 2004-2006 period. The results of the t-tests evidence that for the case of new developments Web projects are not significantly older than non-Web ones. For enhancements and for the total sample non-Web projects are significantly older. This can be a source of bias and poses a limitation on the research results. However, the majority of the prior research papers have not paid attention to this issue.

¡Error! La autoreferencia al marcador no es válida. contains the main descriptive statistics for the effort and for software size indicators. We emphasize that new development projects are bigger than enhancements.

In this Section we show the results of the proposed model. Table 6 provides the parameter estimation for all projects,

Table 5. Effort and Project size: descriptive statistics

Panel A: New developments		
	EFFORT	ADJFP
Mean	3710.647	344.5314
Standard Deviation	6478.468	357.564
25% percentile	965	146
Median	2321	250
75% percentile	4067	372
Minimum	102	21
Maximum	73501	2550
Panel B: Enhancements		
	EFFORT	ADJFP
Mean	1879.911	158.5564
Standard Deviation	4267.063	239.8804
25% percentile	200	43
Median	573	91
75% percentile	1678	173
Minimum	8	4
Maximum	57811	2087
Panel C: Total		
	EFFORT	ADJFP
Mean	2524.405	222.0272
Standard Deviation	5223.204	300.0362
25% percentile	333	63.5
Median	950	138.5
75% percentile	2804	256.5
Minimum	8	4
Maximum	73501	2550

Source: Elaborated by the authors.

Table 6. Total. Equation Estimates

	Coefficient	Std err	t-statistic	p-value
Intercept	6.807844	0.032979	206.43	0.000
b	5.03e-06	7.81e-07	6.44	0.000
c	6.14e-06	8.43e-07	7.28	0.000
K	-0.0000411	4.74e-06	-8.67	0.000
Other statistics				
F test	530.12	P=0.000		
Adj. R ²	0.731			
Cook-Weisberg / Pagan-Breusch	0.140	P=0.705		
Max. CI	9.873			

Source: Elaborated by the authors.

Table 7. Enhancements. Equation Estimates

	Coefficient	Std err	t-statistic	p-value
Intercept	6.78507	0.0447067	151.77	0.000
b	3.86e-06	1.07e-06	3.62	0.000
c	7.24e-06	1.14e-06	6.33	0.000
K	-0.000048	6.16e-06	-7.79	0.000
Other statistics				
F test	334.87	P=0.000		
Adj. R ²	0.725			
Cook-Weisberg / Pagan-Breusch	0.24	P=0.627		
Max. CI	9.631			

Source: Elaborated by the authors.

Table 8. New developments. Equation Estimates

	Coefficient	Std err	t-statistic	p-value
Intercept	6.902051	0.0747149	92.38	0.000
B	7.19e-06	1.18e-06	6.10	0.000
C	2.98e-06	1.44e-06	2.07	0.040
K	-0.0000215	8.42e-06	-2.55	0.011
Other statistics				
F test	76.08	P=0.000		
Adj. R ²	0.5223			
Cook-Weisberg / Pagan-Breusch	0.20	P=0.652		
Max. CI	12.099			

Source: Elaborated by the authors.

while Tables 7 and 8 show the results for enhancements and new developments, respectively. In each of the tables we provide the parameter estimates of the equation, the adjusted R², the results of the F-test and the Cook-Weisberg / Pagan-Breusch test for heteroskedasticity and the maximum of the condition indices (Max. CI), which we used to test the multicollinearity of the models.

7. Discussion

It is noticeable that none of the models show a significant amount of heteroskedasticity or multicollinearity. According to the Cook-Weisberg / Pagan-Breusch test, heteroskedasticity is rejected in all cases. Condition indices

are always below 15, which, according to Netter et al. [49], is a sensible threshold for multicollinearity.

With regard to the parameter estimation, it is evidenced that consistently across all samples, c and K are, respectively, significantly negative and positive. This suggests that both H1 and H2 hold. That is, for the case of Web projects the diseconomies of scale are higher than for non-Web ones. Also, the specific features of Web developments have an influence on the aggregate effect of the cost drivers. This makes the cost of a Web project to be significantly lower than that of a non-Web project with the same features (cost drivers).

Then, we can assert that the development of effort prediction models which are specifically designed for Web applications is clearly justified. Another important issue which stems from our results is that the features of a project have an influence on its cost patterns, both on the intensity of the diseconomies of scale and on the influence of each one of the cost drivers on the total cost of the project.

Our results suggest that cost estimation in software engineering projects requires that projects are grouped with similar characteristics. In the present research the evidence that diseconomies of scale are greater in web projects is shown. Nevertheless, this result is unobservable when the costs of all projects (without splitting) are considered and therefore the project costs of Web developments may be underestimated. This could be the reason for high cost overrun that Hu et al. indicate [18].

We should note that in this paper the limitations of the ISBSG Repository are present. These limitations are common to other papers as Fernández-González and Ladrón de Guevara [50] have shown: (i) in ISBSG the best projects have been selected and the dataset is likely subject to biases, (ii) the ISBSG data is collected from various worldwide organizations with dissimilar backgrounds, business cultures, levels of personnel experience, and development maturity, (iii) a multi-company dataset such as ISBSG also suffers from the presence of more outliers in comparison to a single-company dataset.

8. Conclusions

In the present research we tested the hypothesis of whether the cost behavior of Web developments is significantly different. We focused on two aspects of effort cost behavior: the diseconomies of scale and the influence on the cost of a series of cost drivers that describe the main features of a software project.

With regard to the diseconomies of scale, we hypothesized that they are higher for Web projects due to the relevance of the coordination issue for these kind of projects. Regarding the influence on the effect of the cost drivers, we hypothesized that on an aggregate basis the Web nature of a project mitigates the cost-increasing effects of cost drivers, mainly due to the more frequent use of agile development methods.

We tested our hypotheses on a huge database containing data from more than 5000 projects. This allowed us to apply a series of filters in order to achieve a high degree of homogeneity in the final database. The factors we controlled

for were the quality of the data, the year of the completion of the project and the approach used for measuring the software size. These two last factors have seldom been taken into account in prior research efforts.

Our results indicate that our hypotheses hold. For Web projects, diseconomies of scale are significantly higher and the cost-increasing effects of the multipliers which act as cost drivers are significantly lower. Thus, the main contribution of the present research to the literature is that we provide evidence suggesting that the development of effort prediction models specific for Web applications is justified.

Our findings also raise some questions that may constitute further avenues of research. First, due to data limitations we have studied the effect of the Web nature of projects on the aggregate behavior of the cost drivers. Through a detailed survey on one or more organizations, detailed information on COSYSMO parameters could be gathered for a number of projects significant enough to conduct empirical research. This would allow us to test the influence of the Web nature of software projects on each one of the cost drivers.

Second, the influence on the effort costs of other project features like the type of organization (banking and financial organizations, manufacturing companies, etc.) or the function for which the software product is developed (e.g. accounting, stock control, management information system, etc.) could also deserve detailed study.

Acknowledgments

This work has been funded by the European Union, through the European Regional Development Funds (ERDF); and the Principality of Asturias, through its Science, Technology and Innovation Plan (grant GRUPIN14-100).

References

- [1] Shepperd, M., Cost prediction and software project management. In: Ruhe, G, Wohlin, C, eds. *Softw. Proj. Manag. a Chang. World*, Berlin, Heidelberg: Springer Berlin Heidelberg; 2014, pp. 51–71. DOI:10.1007/978-3-642-55035-5.
- [2] Morisio, M., Stamelos, I., Spahos, V., Romano, D. and Torino, P., Measuring functionality and productivity in web-based applications : a Case study. *Proc. Sixth Int. Softw. Metrics Symp. (Cat. No.PR00403)*, IEEE Comput. Soc; pp. 111–118. 1999. DOI:10.1109/METRIC.1999.809732.
- [3] Fewster, R., Mendes, E., Measurement, prediction and risk analysis for Web applications. *Proc. Seventh Int. Softw. Metrics Symp., IEEE Comput. Soc; 2001*, pp. 338–348. DOI:10.1109/METRIC.2001.915541.
- [4] Mendes, E., Counsell, S. and Mosley, N., Measurement and effort prediction for Web applications. In: Murugesan, S. and Deshpande, Y., eds. *Web Eng.*, vol. 2016, Berlin, Heidelberg: Springer, Berlin, Heidelberg, pp. 295–310, 2001. DOI:10.1007/3-540-45144-7.
- [5] Mendes, E., Mosley, N. and Counsell, S., Web metrics - estimating design and authoring effort. *IEEE Multimed*, 8, pp. 50–57. 2001. DOI:10.1109/93.923953.
- [6] Mendes, E., Mosley, N. and Counsell, S., The application of case-based reasoning to early Web project cost estimation. *Proc 26th Annual Int Comput Softw Appl.* pp. 393–398, 2002. DOI:10.1109/CMPSAC.2002.1045034.
- [7] Mendes, E., Watson, I., Triggs, C., Mosley, N., Counsell, S., Street, S., et al., A comparison of development effort estimation techniques for Web hypermedia applications. *Proc. Eighth IEEE Symp. Softw. Metrics, IEEE Comput. Soc;* pp. 131–140, 2002. DOI:10.1109/METRIC.2002.1011332.

- [8] Mendes, E., Mosley, N. and Counsell, S., Do adaptation rules improve web cost estimation?, Proc. fourteenth ACM Conf. Hypertext hypermedia - HYPERTEXT '03, New York, New York, USA: ACM Press, 2003, 173 P., DOI:10.1145/900051.900091.
- [9] Baresi, L., Morasca, S. and Paolini, P., Estimating the design effort of Web applications. Proceedings. 5th Int. Work. Enterp. Netw. Comput. Healthc. Ind. (IEEE Cat. No.03EX717), IEEE Comput. Soc; pp. 62–72, 2003. DOI:10.1109/METRIC.2003.1232456.
- [10] Ruhe, M., Jeffery, R. and Wieczorek, I., Cost estimation for Web applications. 25th Int. Conf. Softw. Eng. 2003. Proceedings, IEEE; 6, pp. 285–294, 2003. DOI:10.1109/ICSE.2003.1201208.
- [11] Ruhe, M., Jeffery, R. and Wieczorek, I., Cost estimation for Web applications. 25th Int. Conf. Softw. Eng. 2003. Proceedings., IEEE, pp. 285–294, 2003. DOI:10.1109/ICSE.2003.1201208.
- [12] Srivastava, S.K., Prasad, P. and Varma, S.P. Evolving predictor variable estimation model for Web engineering projects. In: Rajesh, R. and Ranjan, P., eds. Comput. Sci. Eng., New Delhi: Narosa Publishing House, 2015, pp. 68–89.
- [13] Reifer, D.J., Web development: Estimating quick-to-market software. IEEE Softw., 17, pp. 57–64, 2000. DOI:10.1109/52.895169.
- [14] Ruhe, M., Jeffery, R. and Wieczorek, I., Using Web objects for estimating software development effort for Web applications. Proceedings 5th Int Work Enterp Netw Comput Healthc Ind (IEEE Cat No03EX717), pp. 30–7, 2003, DOI:10.1109/METRIC.2003.1232453.
- [15] Mendes, E. and Mosley, N., Bayesian network models for Web Effort prediction: A comparative study. IEEE Trans Softw Eng., 34, pp.723–737, 2008. DOI:10.1109/TSE.2008.64.
- [16] Mendes, E., A comparative study of cost estimation models for Web hypermedia applications. Empir Softw Eng., 8, pp.163–196. 2003. DOI:10.1023/A:1023062629183.
- [17] Lagerström, R., von Würtemberg, L.M., Holm, H. and Luczak, O., Identifying factors affecting software development cost and productivity. Softw Qual Journal, 20, pp. 395–417, 2011. DOI:10.1007/s11219-011-9137-8.
- [18] Hu, Q., Plant, R.T. and Hertz, D.B., Software cost estimation using economic production models. J. Manag Inf Syst., 15, pp. 143–163, 1998.
- [19] Sankhwar, S. and Pandey, D., Software project risk analysis and assessment: A survey. Global Journal of Multidisciplinary Studies, 3 (5), pp. 144–160, 2014.
- [20] Mendes, E., Mosley, N. and Counsell, S., Early Web size measures and effort prediction for Web costimation. Proceedings 5th Int Work Enterp Netw Comput Healthc Ind (IEEE Cat No03EX717), 2003, pp. 18–39. DOI:10.1109/METRIC.2003.1232452.
- [21] Mendes, E. and Mosley, N., Further investigation into the use of CBR and stepwise regression to predict development effort for Web hypermedia applications. Proc. Int. Symp. Empir. Softw. Eng., 2002, pp. 79–90. DOI:10.1109/ISESE.2002.1166928.
- [22] Pressman, R.S., What a tangled Web we weave [Web engineering]. IEEE Software., 17 (1), pp.18–21, 2000. DOI:10.1109/52.819962.
- [23] Reifer, D., Ten deadly risks in Internet and intranet software development. IEEE Software, 19 (2), pp. 12–14. 2002. DOI:10.1109/52.991324.
- [24] Ochoa, S.F.S.F., Bastarrica, M.C.C., Parra, G., Estimating the development effort of Web projects in Chile, in: Web Congress, 2003. Proceedings. First Latin American, pp. 114–122, 2003. DOI:10.1109/LAWEB.2003.1250289.
- [25] Ambler, S.W., Lessons in agility from Internet-based development. IEEE Software, 19 (2), pp. 66–73, 2002. DOI:10.1109/52.991334.
- [26] Pardo, C., Hurtado, J.A. and Collazos, C.A., Agil software process improvement with agile spi - process, DYNA, 77 (164), pp. 251–263, 2010.
- [27] Beck, K., Extreme programming explained: Embrace change. addison-wesley professional, 2000.
- [28] (Dick)-Fairley, R.E. The influence of COCOMO on software engineering education and training. Journal of Systems and Software, 80 (7), pp. 1201–1208. 2007. DOI:10.1016/j.jss.2006.09.044.
- [29] Boehm, B.W., Software engineering economics. 1st ed. Prentice Hall; 1981.
- [30] Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R. and Selby, R., Cost models for future software life cycle processes: COCOMO 2.0. Ann Softw Eng., 1, pp. 57–94, 1995. DOI:10.1007/BF02249046.
- [31] Patil, L.V., Waghmode, R.M., Joshi, S.D. and Khanna, V., Generic model of software cost estimation: A hybrid approach. 2014 IEEE Int. Adv. Comput. Conf., IEEE, pp. 1379–1384, 2014. DOI:10.1109/IAAdCC.2014.6779528.
- [32] Patil, P.K., A review on calibration factors in empirical software cost estimation (SCE) models. Int. J. Softw. Eng. Res. Pract., 3, pp. 1–7, 2014.
- [33] Paulk, N.C., Extreme programming from a CMM perspective. IEEE Softw., 18 (6), pp. 19–26, 2001. DOI:10.1109/52.965798.
- [34] Schwaber, K. and Beedle, M., Agile software development with scrum. Prentice Hall, 2002.
- [35] Reichtin, E., Systems architecting: Creating and building complex systems, Prentice Hall, 1991.
- [36] Valerdi, R., The constructive systems engineering cost model (Cosysmo): Quantifying the costs of systems engineering effort in complex systems, VDM Publishing, 2008.
- [37] Valerdi, R., Heuristics for systems engineering cost estimation. IEEE Syst. J., 5, pp. 91–98, 2011. DOI:10.1109/JSYST.2010.2065131.
- [38] Coram, M. and Bohner, S., The impact of agile methods on software project management. 12th IEEE Int. Conf. Work. Eng. Comput. Syst., IEEE; pp. 363–370, 2005. DOI:10.1109/ECBS.2005.68.
- [39] Cuadrado, J., Sicilia, M., Garre, M. and Rodriguez, D., An empirical study of process-related attributes in segmented software cost-estimation relationships. Journal of Systems and Software, 73 (3), pp. 353–361, 2006. DOI:10.1016/j.jss.2005.04.040.
- [40] Aroba, J., Cuadrado-Gallego, J.J., Sicilia, M.-Á., Ramos, I. and García-Barriocanal, E., Segmented software cost estimation models based on fuzzy clustering. J. Syst, Softw., 81, pp. 1944–1950, 2008. DOI:10.1016/j.jss.2008.01.016.
- [41] Song, Q. and Shepperd, M., A new imputation method for small software project data sets. J. Syst. Softw., 80, pp. 51–62, 2007. DOI:10.1016/j.jss.2006.05.003.
- [42] Rodríguez, D., Sicilia, M.A., García, E. and Harrison, R., Empirical findings on team size and productivity in software development. J. Syst. Softw., 85, pp. 562–570, 2012. DOI:10.1016/j.jss.2011.09.009.
- [43] Bibi, S., Tsoumakas, G., Stamelos, I. and Vlahavas, I., Regression via classification applied on software defect estimation. Expert. Syst. Appl., 34, pp. 2091–2101, 2008. DOI:10.1016/j.eswa.2007.02.012.
- [44] Heričko, M., Živković, A. and Rozman, I., An approach to optimizing software development team size. Inf. Process Lett., 108, pp. 101106, 2008. DOI:10.1016/j.ipl.2008.04.014.
- [45] Huang, S.-J. and Han, W.-M., Selection priority of process areas based on CMMI continuous representation. Inf. Manag., 43, pp. 297–307. 2006. DOI:10.1016/j.im.2005.08.003.
- [46] Sayyad-Shirabad, J. and Menzies, T.J., The PROMISE Repository of software engineering databases. Thesis, School of Information Technology and Engineering. Univ Ottawa, Canada, 2005. Available at: <http://promise.site.uottawa.ca/SERepository>.
- [47] Pendharkar, P.C. and Rodger, J.A., Subramanian GH. An empirical study of the Cobb–Douglas production function properties of software development effort. Inf. Softw. Technol., 50, pp. 1181–1188, 2008. DOI:10.1016/j.infsof.2007.10.019.
- [48] Vogelegang, F. and Lesterhuis A., Applicability of COSMIC full function points in an administrative environment: Experiences of an early adopter. Proc. 13th Int. Work. Softw. Meas. – IWSM, Verlag, pp. 23–25, 2003.
- [49] Kutner, M., Nachtsheim, C., Neter, J. and Li, W., Applied linear statistical models, Vol. 1., 5th ed.. McGraw-Hill/Irwin, 2004.
- [50] Fernández-Diego, M. and González-Ladrón-de-Guevara, F., Potential and limitations of the ISBSG dataset in enhancing software engineering research: A mapping review. Inf. Softw. Technol., 56, pp. 527–544, 2014. DOI:10.1016/j.infsof.2014.01.003.

J. de Andrés, received his BBA in 1993 and his PhD in 1998, both of them from the Universidad de Oviedo, Spain. He is an associate professor of Accounting and Finance at the University of Oviedo, Spain. His research interests include artificial intelligence applied to business organizations and technology management. Dr. De Andrés has published over 60 articles in

journals and refereed proceedings. He is Vice Dean of the School of Computer Science of the University of Oviedo.
ORCID: orcid.org/0000-0001-6887-4087

D. Fernández-Lanvín, received his BSc. in Computing in 1998, his MSc. in Computing in 2002, and his PhD degree in 2007, all of them from the Universidad de Oviedo, Spain. From 1998 to 2003 he worked as software engineer in several companies, developing mainly web applications. Since 2003 he has worked as an associate professor in the Department of Computing of the University of Oviedo, Spain. His research interests are Web development, HCI and project management.
ORCID: orcid.org/0000-0002-5666-9809

P. Lorca, received his BBA in 1993 and his PhD in 2000, both from the Universidad de Oviedo, Spain. He is an associate professor of accounting and finance at the University of Oviedo, Spain. His research interests include ERP systems and technology management. Dr. Lorca has published over 50 articles in journals and refereed proceedings. He is a member of the Standing Committee on New Technologies and Accounting of the Spanish Association of Accounting and Business Administration.
ORCID: orcid.org/0000-0002-7640-8286



UNIVERSIDAD NACIONAL DE COLOMBIA

SEDE MEDELLÍN
FACULTAD DE MINAS

Área Curricular de Ingeniería
de Sistemas e Informática

Oferta de Posgrados

Especialización en Sistemas
Especialización en Mercados de Energía
Maestría en Ingeniería - Ingeniería de Sistemas
Doctorado en Ingeniería- Sistema e Informática

Mayor información:

E-mail: acsei_med@unal.edu.co
Teléfono: (57-4) 425 5365