Estimating the Development Effort of Web Projects in Chile

Sergio F. Ochoa
Computer Sciences Department
University of Chile
(56 2) 678-4364
sochoa@dcc.uchile.cl

M. Cecilia Bastarrica
Computer Sciences Department
University of Chile
(56 2) 678-4364
cecilia@dcc.uchile.cl

Germán Parra
Computer Sciences Department
University of Chile
(56 2) 678-4364
gparra@dcc.uchile.cl

Abstract

This article presents a method to fast estimate the development effort of Web-based information systems in Chile. The method, called Chilean Web Application Development Effort Estimation (CWADEE), addresses a necessity to get effort estimations in a short period of 24 to 72 hours using limited information. In contrast with other existing methods, CWADEE uses raw historical information about development capability and high granularity information about the system to be developed, in order to carry out such estimations. This method is simple and specially suited for small or medium-size Web-based information systems. CWADEE has been applied to twenty-two projects with very accurate results.

Keywords

Effort Estimation Method, Sizing Metric, Web-based Information Systems, Web Engineering.

1. Introduction

Companies developing Web-based systems face the challenge of estimating the required development effort in a very short time frame. This problem does not have a standard solution yet. On the other hand, effort estimation models that have been used for many years in traditional software development are not very accurate for Web-based software development effort estimation [16].

Web-based projects are naturally short and intensive [12], so not having an appropriate effort estimation model pushes developers to make highly risky estimations [16]. Moreover, the rapid evolution and growth of Web related technology, tools and methodologies makes historical information quickly obsolete. Although, the software effort estimation process is a completely necessary and critical task, it still looks more like a craft than a science [4][17]. The process is mainly dependent of the project type and the features of the development scenario.

This work is focused on small and medium sized Web-based information systems developed in immature scenarios. This kind of projects represents about 50% of the projects that are being developed in Chile today [18]. Chilean developers face a highly competitive market, where clients require cost estimations in a matter of hours for their projects, and these estimations should be accurate if they want the development to be cost effective. In large projects, the estimation is less problematic because there is always time scheduled for carrying out this process. These estimations are the basis of the budget given to the client. Based on such budget the software development companies sign contracts with the client. In other words, the effort estimation carried out for budget purposes usually establishes the business rules for the project.

Without an appropriate model, cost estimation is done with a high uncertainty and the development effort estimation relies only on the experience of an expert, whose estimations are generally not formally documented. This expert knows well the development capabilities of the company and is able to interpret the client's requirements with high accuracy. Chilean development companies rarely change the expert in charge of effort estimation, in contrast to what occurs with other personnel involved in Web-based software development. This person usually becomes a bottleneck for budget delivering. Therefore, the estimation process can be slow and/or little accurate. This is a critical situation for development. Chilean clients are "budget centric", being such budget the base of the contract.

In order to deal with such problem we have been studying the last 3 years the software development process in Chile, oriented to the development of small and medium size Web-based information systems. Based on the analysis of these results, we identified a low usability of the well-known effort estimation methods and a necessity of a model to support estimation in such scenario. Due that, we developed a method for fast estimating the Web-based software development effort and duration, which is adapted to development of Web-based projects in Chile. We called it Chilean Web Application Development Effort Estimation (CWADEE). The method is specifically applicable to estimate the development effort of small to medium-size Web-based information systems in immature development scenarios.

Proceedings of the First Latin American Web Congress (LA-WEB 2003)

It is probable that this development scenario is similar to those in other Latin America's countries, although currently we cannot assure it.

Local traditions are difficult to change. CWADEE does not replace the expert in charge of effort estimation process. Instead, it provides him/her with a tool for making more precise estimations and based on more sound bases in the short time available for such task. Based on our previous studies and knowledge of the Chilean scenario, we estimate that the current error range in development effort estimations for small and medium size projects is between 50 and 200%. The proposed method intends to support the experts in order to reduce this range.

Next Section presents a more detailed description of the Chilean effort estimation scenario. Section 3 discusses the work done in software development effort estimation and methods and tools specifically for Web-based systems. Section 4 describes the CWADEE method and its main components. Section 5 presents and analyzes the results obtained in our experience applying it. Finally, Section 6 presents the conclusions and the future work.

2. Effort Estimation Scenario

The Chilean effort estimation scenario is not completely different from other scenarios, but due to some of its characteristics, well-known effort estimation methods have low applicability. The central characteristics are the following:

Expert centered. An expert is in charge of the effort estimation process, who knows the development capability of the organization. He/she is responsible for calculating the budget for clients and to establish development compromises. The expert's experience is very useful for the company but not necessarily good for other companies. Companies rarely risk recruiting new people for this task because the learning cost of the expert is high. Because of this feature, the amount of experts in each development company is stable and usually quite low trying not to increase the fix costs of the company. These experts are expensive due to the responsibility they have.

Little historical information. Development companies generally have little historical information about past projects. Usually, they have the products produced by the project and an approximation of the total amount of man-hour spent on it. Besides, such information is usually unorganized and it is perceived by the expert as not very reliable because of its incompleteness.

Short time to estimate. Development companies generally have between 24 and 72 hours to estimate the development effort of small or medium sized projects, from the moment the client provides the Web application requirements and information. In that time, the experts should analyze and clarify the information provided by the client, revise historical information, carry out the estimation, and build the budget.

Gross gained information. The information given by the client about the problem to be solved tends to be gross grained and could have unclear areas for the client. The client wants the estimation to include some flexibility to adjust such unclear issues. Therefore, a lot of experience is required from the estimator to dimension the system realistically only based on high level information.

Quite fixed development time. Generally, the development projects arise as response to immediate needs of the clients. Therefore, the development time for a project is quite fixed and the estimation process becomes a problem of feasibility and/or money.

Budget centric. The Chilean scenario is "budget centric". Typically, the clients request budgets to several development companies and usually the lowest development cost is chosen. This means that the developer should prepare budgets for many clients which could make the experts to become a bottlenecks. On the other hand, work of such experts is an investment with low probability of success, because several budgets are requested and only one is chosen.

These features make the Chilean effort estimation scenario be highly demanding and little motivating. The well-known effort estimation models have shown low applicability to support this process; therefore, it is currently carried out in a handmade way. Next Section, presents and discusses some related work.

3. Related Work

From the beginning of software engineering as a research area more than three decades ago, several development effort estimation methods have been proposed. We can classify these methods for our research as those for *traditional software* and those for *Web-oriented software*. The traditional effort estimation methods are those used to estimate the development effort of software that consists of programs in a programming language, which eventually interact with data files or databases. Generally, these software have an active execution thread that provides system services. On the other hand, the Web-oriented methods use different metrics and they are focused on estimating the development effort of products that are event-oriented. These products generally involve code in a programming language, imagery, look-and-feel, information structure, navigation and multimedia objects.

Traditional effort estimation methods like COCOMO [1][3] are mainly based on metrics like Lines Of Code (LOC) [15] or Function Points (FP) [10][13]. The estimation strategies supported by LOCs have shown several problems. Most working Web projects agree that LOCs are not suitable for early estimation because they are based on design [16]. Other reported problem is that the work involved in the development of multimedia objects and look-and-feel cannot be measured in LOCs. Also, an important amount of reliable historical information is needed to estimate effort using this metric, and this information is hard to get in Web-based projects. Finally, to carry out estimations using LOCs requires a long analysis of the historical information, which reduces the capability to get reliable fast estimations. Speed is an important requisite of Web-based projects developed in Chile.

Similarly, traditional estimation methods based on FPs are not appropriate because applications do more than transform inputs to outputs, i.e. the effort necessary for developing a Web-based application is much more than the effort required for implementing its functionality. FPs do not consider the imagery, navigation design, look-and-feel, and multimedia objects, among others. In other words, the traditional categories of FPs should be redefined. This kind of estimation also requires an important amount of reliable historical information, which supports the used values of each FPs.

Although there are several software effort estimation methods like Price-S, Slim and Seer [3], COCOMO is the most well known and used by the software industry. It had shown to be appropriate in many development scenarios. The first version of such method used LOCs as the fundamental metric to support the estimations. Then, Boehm proposed COCOMO II, which could use alternatively LOCs, FPs or Object Points [2]. Although COCOMO II was not defined to support the development effort estimation of Web applications, many people found the way to adapt the object point concept in order to get a sizing estimation [3]. Object points are an indirect metric, similar to FPs, which considers three categories: user interfaces, reports and components, which are probably needed to develop the final product. Every element in the system is categorized and classified in three complexity levels: basic, intermediate and advanced. Then, based on these classified elements, and taking into account the historical information, it is possible to generate a good estimation. Object Points and COCOMO II seem to be acceptable for traditional or multimedia software projects, but they are not good enough to get accurate effort estimations for Web-based information systems developed in Chile. The complexity of the estimation process and the need for detailed historical information make them difficult to apply in this scenario.

Several size metrics have been proposed for Web applications, like Object Points, Application Points and Multimedia Points [6]. However, the most appropriate seems to be Web Objects (WO) [16]. WOs are an indirect metric that is based on a predefined vocabulary that allows defining Web systems components in term of operands and operators. To estimate the amount of WOs that are part of a Web-based application it is necessary to identify all the operators and operands present in the system. Then, they are categorized using a predefined table of Web Objects predictors and also they are classified in three levels of complexity: low, average and high. The final amount of WO in a Web-based application is computed using the Halstead equation [10], and it is known as the volume or size of the system. The effort estimation and the duration of the development are computed using WebMo (Web Model), which is an extension of COCOMO II [16]. This model uses two constants, two power laws, several cost drivers, and the product size expressed in WO (see Figure 1).

$$\text{Effort} = A\prod_{i=1}^{9} cd_i(\text{Size})^{P1} \qquad \text{Duration} = B(\text{Effort})^{P2}$$

Where: A and B are constants

P1 and P2 are power laws

cd_i are cost drivers

Size is the number of Web Objects

Figure 1. WebMo Effort Estimation Model

Constants A and B, and power laws P1 and P2 are defined by a parameter table in the model. This table contains the values obtained from a database of former projects (historical information). The cost drivers are parameters used to adjust the effort and duration in terms of the development scenario. For this model nine cost drivers were defined: product reliability and complexity (RCPX), platform difficulty (PDIF), personnel capability (PERS), personnel experience (PREX), facilities of tools and equipment (FCIL), scheduling (SCED), reuse (RUSE), teamwork (TEAM) and process efficiency (PEFF) [16]. Each cost driver has different values that may be: very low, low, normal, high, and very high.

The combination of WebMo and Web Objects is, in this moment, the most appropriate method to estimate the development effort of Web applications. However, this combination does not seem to be the best for Chilean development scenarios because it needs an important amount of historical detailed information to carry out the estimation. Also, the WO identification and categorization process is difficult to carry out in a short time, and it requires an expert that also knows how to carry out the project. The expert should be competent about critical technical decisions of the development process because in small projects the technical feasibility study is implicitly included in this estimation [4]. This expert feature and the complexity to identify and classify WO make WebMo unfeasible to use in Chilean scenarios.

Provided that the effort estimation methods presented are no appropriate to estimate the development effort of Web-based information systems in Chilean scenarios, in the next section we present the CWADEE method. It intends to be more appropriate to estimate the development effort of small or medium size projects, especially in scenarios that require fast estimation with little historical information.

4. The CWADEE Method

In order to help the expert achieve more accurately effort estimations in Chilean scenarios, CWADEE introduces a new sizing metric based on the data model of the information system to be developed: Data Web Points (DWP). DWP is an indirect sizing metric that takes into account the characteristics of Chilean experts, the available estimation time, the lack of a great amount of historical information, and the kind of information (high level) used to estimate. The idea behind the DWP is to identify the system functionality, analyzing its data model. Trujillo and Straub showed it is possible to infer between 50 to 60 percent of an information system functionality following a set of predefined patterns [20]. In this way, DWP capture the essence of these patterns making explicit the relationship between the data model and the size of the Web application. Each pattern corresponds to one and only one kind of DWP, and vice versa. Information systems functionality is largely determined by the data model. Generally, these systems are organized as a combination of repository and MVC patterns [5]. DWP takes advantage of the standard relationship between data model and functionality in an information system. Next section describes the concept of Data Web Point. Section 4.2 presents the CWADEE effort estimation method. Then, the following sections explain the components, which are part of CWADEE.

4.1 The Data Web Points

DWPs are similar to other indirect metrics such as FPs [10][13], Object Points [2, 3], or Web Objects [16] in the fact that they represent abstract concepts that are used to obtain the size of the system to be developed. In particular, DWPs represent system functionality from the point of view of its data model. Therefore, two experts analyzing the same data model should get the same size for a system, but the development effort could be different depending on particular characteristics of the development environment such as: the expertise of the development team, and the available supporting tools, among others.

DWP intends to simplify the effort estimation process by giving the expert better tools for improving or tuning up this process. This metric has the advantages of being fast and easy to apply, whenever the required gross grain historical information is available. This fact makes it especially suitable to estimate Web-based projects in Chile.

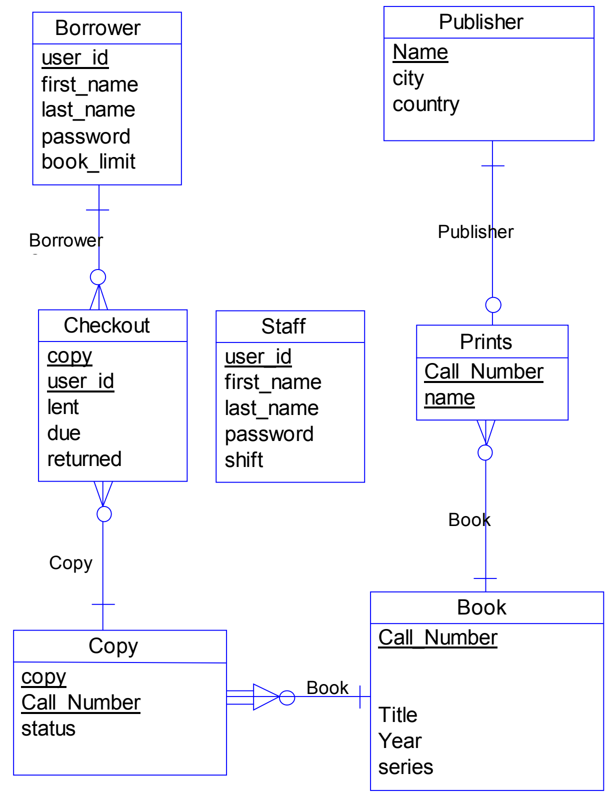

Figure 2. Example of an Information System of a Library

The types of entities and relationships associated to DWPs are the following: *regular*, *dependent*, *relation*, *relationship 1 to N* and *relationship 1 to 1*. Regular entities are those that do not contain a foreign key and whose primary key identifies them (*Borrower, Staff, Book* and *Publisher* in Figure 2). Dependent entities are those related to a regular entity through a foreign key and whose primary key is formed by the primary key of a regular entity and a key of its own (*Copy*). Relation entities separate a N to M relationship between two other entities and they do not have a primary key of their own; its key is formed by the primary keys of the entities participating in the relationship (*Checkout* and *Print*). This type of entity can also have other attributes defined. Relationships 1 to N are established between a relation entity and another entity, where one or more instances of the relation entity are associated with an instance of the other entity (*Borrower, Checkout, Copy Checkout* and *Book Print*). Finally, relationships 1 to 1 are those established between a relation entity and another one, where one and only one instance of the relation entity is associated with an instance of the other entity (*Publisher Print*).

Taking into account the patterns defined in [20], the different categories of DWPs follow these patterns. In contrast to Fowler's analysis patterns [7] and Hay's data model patterns [9], here each type of entity and relationship that represents a pattern is considered a

Proceedings of the First Latin American Web Congress (LA-WEB 2003)
0-7695-2058-8/03 $17.00 © 2003 IEEE

DWP. In order to facilitate the explanation of DWPs, we will use a simple library data model, shown in figure 2.

Each of these entities represents a design pattern that has a typical functionality associated with it. This functionality can be translated into standard services provided to Web-based information systems. Trujillo has shown that not only services, but also the strategy of using the data model are useful as a basis for developing information systems in the Chilean scenario [20].

Table 1. Definition of the DWPs amount

Type of DWP	Amount of DWP		Weight Factor		Total of DWP
Regular Entities	4	x	9	=	36
Dependent Entities	1	x	9	=	9
Relationship Entities	2	x	3	=	6
Relationship 1 a N	3	x	6	=	18
Relationship 1 a 1	1	x	3	=	3
Total of DWP	→			=	72

Analyzing the data model we can easily identify DWPs through of a systematic process; so systematic that it could even be automated. Thus, we can fill the table of DWPs in Table 1 to calculate the system size. The weight assigned to each category of DWP represents the development effort of each one, and it is based on the experience of the expert estimator. The expert can update this weight every time he/she believes it is appropriate. Table 1 shows the estimation of the DWPs associated with the model shown in

Figure 2. The resulting size value will be considered as the main factor in the CWADEE effort estimation equation (in section 4.2).

This metric seems to be more usable for expert estimators in Chile, provided that most of them could apply the proposed data model analysis. The gross grain information obtained in interviews with the client usually allows us to build a good and fast approximation to the final data model. Then, the DWP can be directly derived from this model. Up to this point all the process is systematic and unambiguous. The knowledge of the expert is required for more challenging activities such as: defining appropriate cost drivers and weights for DWPs. Generally, these tuning aspects have direct incidence on the accuracy of the effort estimation for this kind of projects.

4.2 Effort Estimation

The DWPs are an approximation of the whole size of the project; so, it is necessary to know what portion of the whole system DWPs represent. This knowledge is achieved through a relatively simple process (briefly described in Section 4.3) and it is generally done by the expert estimator. Assuming that the estimation factors in the computation of the effort are subjective, flexible and adjustable for each project, the role of the expert becomes very relevant. Once the value of the portion or representativeness is calculated, the expert can adjust the total number of DWPs and he/she can calculate the development effort using the following equation.

$$E = UC \cdot \prod_i cd_i \cdot \left(DWP \cdot (1 + X^*)\right)^P$$

Where:

E is the development effort measured in man-hours.

UC is the *User Cost*.

cd_i are each of the *Cost Drivers*.

DWP corresponds to the Web application size in terms of Data Web Points.

X^* is the coefficient of DWP representativeness.

P is a constant.

The estimated value of real Data Web Points (DWP*) is calculated as the product of the initial DWPs and the representativeness coefficient X^*. This coefficient is a historical value that indicates the portion of the final product functionality that cannot be inferred from the system data model. The process of defining such coefficient is presented in the next section. The user cost is described in Section 4.3, and it represents the system functionality that is associated with each user type. The defined cost drivers (cd_i) are defined in Section 4.4, and they are similar to those defined by Reifer for WebMo [16]. The last adjustable coefficient in CWADEE corresponds to constant P that is the exponent value of the DWP*. This exponent is a value very close to 1.00, and it must neither be higher than 1.10 nor lower than 0.99. This constant's value depends on the project size measured in DWPs. In order to determine this value, 22 Web-based projects developed in the Computer Science Department at the University of Chile were studied. As a result, this constant was assigned the value 1.05 for projects smaller than 300 DWPs, and 1.02 for projects larger than 300 DWPs.

4.3 Identification of the Representativeness Coefficient

The representativeness coefficient indicates the amount of additional effort could be needed to address the issues that cannot be inferred from data model, taking into account the history of similar projects. Among these issues are: additional functionality, imagery, look-and-feel, multimedia objects, navigation, and information structure, among others. This coefficient can evolve along the time and it is referential; therefore the expert can adjust it for each project to get more accurate effort estimations.

In order to calculate the representativeness of DWPs, we should analyze similar finished projects, taking into account the system data model and the total man-hours for each project. Using the inverse of the CWADEE equation, we can obtain the value of DWP* (real Data Web Points) used in each project. Such equation is the following:

$$DWP^{*} = \sqrt[P]{\frac{E}{UC \bullet \prod_{i} cd_{i}}}$$

Obtaining the real size in Data Web Points (DWP*) of a finished project, we only require the total man-hours used in it, and the experience of the estimator who can obtain the value of the UC and the *cost drivers.* For this task not much historical information is needed. Once we have the value of DWP* for a historic project, we should compare it with the DWPs that can be obtained from the data model of such project. Thus, it is possible to obtain the representativeness coefficient of the DWPs for the same specific project using the following equation:

$$X = \frac{DWP}{DWP^{*}}$$

Where:

X corresponds to the percentage of the system size which cannot be inferred from the data model using the DWPs.

This process could be repeated in several projects within the company or development team in order to get a more accurate coefficient. The participation of the expert in the estimation is crucial for this reengineering process, because only his/her experience can determine the user cost and the value of the cost drivers that existed in those projects. Thus, we can calculate the weighted mean of the representativeness of the DWPs in the company's history with the following equation:

$$X^{*} = \frac{\sum_{i} E_{i} \cdot X_{i}}{\sum_{i} E_{i}}$$

Where:

E_i is the effort in man-hours used in each project.

X_i is the fraction of the size represented by the DWPs.

X^* is the weighted mean of the fraction of the real size of the project that is calculated from the DWPs of the data models that the company generates.

4.4 User Cost

The *User Cost* (UC) is a function of the user types to be supported by the system. CWADEE considers three user types defined as: manager, updaters and consultants. The manager user is in charge of supervising the available applications in the system, activating and deactivating functional areas of the system, and maintaining the set of applications that keep the project in constant execution. The updater user uses the available functionality in the system to modify and consult to the stored information. Finally, the consultant user has access to part of the information available in the system, but only for reading. On the other hand, CWADEE also considers the possibility that *variable users*, which are a mix of the before mentioned user types, are present in the system. These users are a mix of the aforementioned.

The user types a system supports are important to calculate the development effort because each type requires interfaces developed and adjusted for it, but generally part of the functionality is shared. For this reason, CWADEE considers the scope of each user type in the system, and the functionality reuse among user types in order to calculate the user cost. The *User Cost* is a variable dependent on each project, and so it can be determined in particular for each one, and it is subject to the estimator subjectivity. For calculating the user cost we should build a table similar to the one shown in Table 2.

This table indicates that the three predefined user types are present in the system, where the administrator has access to 30% of the applications functionality; the updater uses 60% and the consultant 80%. There are also in the example two variable user types that correspond to the secretary that uses 30% of the applications functionality and the area manager that uses 20%. On the other hand, each user type has a degree of reuse of system functionality that is determined by the expert estimator. This reuse can come from the implementation of functionality for other user types, or other systems. Once this user types table is generated, it is possible to calculate the *UC* value using the following equation:

$$UC = \sum_{i} I_{i} \cdot \left(1 - R_{i}\right)$$

Where:

UC represents the *User Cost* used in the effort equation.

I_i is the fraction of the scope of each user type.

R_i is the reuse degree of each user type.

This equation yields values between 0 and 5. A value of UC of 0 means the system reuses all the functionality associated with each user type; so, the development effort will also be zero. On the other hand, if the user cost is five, this means that there is no reuse of any kind to implement the system functionality for each user type.

4.5 Cost Drivers

Finally, the CWADEE method has a series of *Cost Drivers* taken from the WebMo model proposed by Reifer [16] These *Cost Drivers* represent the available development scenarios for a particular project. Such scenarios have positive and negative influences over the development process that need to be taken into account during the estimation process. *Cost Drivers* are subjective factors in CWADEE, and all of them, but CLIEN, were defined for the WebMo model.

COPLX: Product reliability and complexity. (Product attributes).

DIFPLA: Platform difficulty. (Platform and net servers volatility).

PERS: Personnel capabilities. (Skills, knowledge and abilities of the work force).

EXPER: Experience of the personnel. (Depth and width of the work force experience).

INFRA: Infrastructure. (Tools, equipment and geographical distribution).

SCHED: Scheduling. (Risk degree assumed if the delivery time is shortened.)

CLIEN: Client type. (Technology knowledge the client has; requirements stability.)

WTEAM: Work team. (Ability to work synergistically as a team).

EFPRO: Process efficiency. (Development process efficiency).

Table 2. Example of a scope table for different user types in a Web-based project

	User types	Fraction of the Scope (I)	Reuse Degree (R)
Fixed Users	Manager	0.3	0.1
	Updater	0.6	0.5
	Consultant	0.8	0.8
Variable Users	Secretary	0.3	1.0
	Area Manager	0.2	0.9

Table 3. Reference values for cost driver

Cost drivers for CWADEE					
Driver	**VL**	**L**	**N**	**H**	**VH**
COPLX	0.65	0.85	1.00	1.35	1.60
DIFPLA	0.90	1.00	1.10	1.30	1.65
PERS	1.60	1.30	1.05	0.90	0.80
EXPER	1.30	1.15	1.05	0.90	0.80
INFRA	1.35	1.15	1.00	0.90	0.85
SCHED	1.40	1.20	1.00	0.95	0.90
CLIEN	1.55	1.35	1.10	0.90	0.75
WTEAM	1.50	1.30	1.05	0.90	0.80
EFPRO	1.30	1.10	1.00	0.85	0.65

Each of these cost drivers is classified in a five level scale: very low, low, normal, high and very high (VL, L, N, H, VH). In order to determine which level corresponds to each cost driver, the estimator uses a series of predefined tables that were built using historical information (gross grain) of other Chilean projects. Each cost driver has an assigned value in each category, and the product of each value is part of the equation for calculating the effort in the CWADEE method. The assigned values in each category are replaced in the CWADEE effort estimation equation in order to obtain the result in man-hours. Using the information of 22 Web-based projects developed by undergraduate and graduate students at the Computer Science Department at the

University of Chile, we have obtained approximate values for each cost driver in each category. Table 3 shows these values.

5. Obtained Results

Computer science graduate and undergraduate students in the regular "Software Engineering" course have used the current version of the CWADEE model for the last 3 years. They used it for calculating the effort required for each project, and to help them establish the limits of the work to be done. At the beginning of the course they are instructed about this effort estimation technique, and the historical cost drivers are presented and discussed. Then, the students are grouped in teams of 5 to 7 people, and roles are assigned to the members. The roles are project manager, analyst, designer, developer and tester. The last one is a distributed role. Also, real projects and clients are assigned to each group. They have only 16 weeks to develop the project and the first week is used to establish the limits of each development. This activity is a negotiation between the project manager and the client based on development effort estimation (***E***), which can be calculated taking into account the time available, the features of work group members, the group size and the problem to be solved. Also, raw information about these issues collected in past projects can be used. The course instructor and two industry experts, who know very well the course's development scenario, evaluate the established limits. These limits are in direct relationship with the effort they estimate needed to develop the project. A summary of the obtained results about the estimations carried out by the students, is shown in Table 4.

By taking into account the DWP identified by each group and the project limit definition, the size estimation (or established limits) is evaluated as good if it is similar to expert opinion. It is evaluated as medium if the estimation and the expert's opinion are close, or poor if they are not similar.

Table 4. Obtained results using CWADEE

Projects Amount	Size Estimation Quality	Final Product Quality
15	Good	In production – 9 Projects
		Need minor adjust – 6 Projects
5	Medium	Need minor adjust – 2 Projects
		Incomplete – 3 Projects
2	Poor	Incomplete – 2 Projects

Table 4 shows that students with little experience can get good estimations using CWADEE taking into account the expert's opinions. Also, the projects well sized were put in production or needed minor adjustments, and the project poor sized were incomplete. Independently of the expert's opinion, these results show that applicability level of CWADEE to support the effort estimation of small and medium sized Web-based development in Chilean scenarios.

On the other hand, at least a couple of students who were taught CWADEE, have created their Web software development companies, and they are currently using this technique to estimate the development effort. The results they report are better than the above presented, but they were obtained in a no-controlled scenario.

The characteristics of the mentioned course development scenario are similar to most of Chilean development companies. Currently, several students whose participate in the course are working in the industry as developers of such projects and they corroborate this fact. Therefore, the obtained preliminary values may be showing the proposed method is appropriated to support the expert to improve the quality of effort estimation in such scenarios.

6. Conclusions and Future Work

The market evolution forces clients to get involved in very short-term Web projects. Too often the results are incomplete, unreliable and difficult to maintain applications that fail to meet the business needs, and cost more and take longer to develop than expected. Generally, the effort estimation is not able to foresee and help avoid these problems. Although developers spend time trying to estimate the software development effort realistically and reliably, they usually have very little time for this task and very little historical information is available. These characteristics tend to make estimations less reliable regarding both time and cost. An expert knows the development scenario and the development capabilities of his/her organization, but he/she generally does not have good tools to support an accurate, reliable and fast estimation (24-72 hours). In this scenario, it is really difficult to fix a competitive budget with little risk of loosing money.

In order to get fast and reliable effort estimations of Web-based information systems development projects, this paper presented the CWADEE method. It does not replace the expert estimator, but it provides him/her with a tool for achieving a more accurate estimation, based on real data in a shorter time. CWADEE has been applied in a development scenario similar to most of the Chilean software development companies and it has shown be useful to support the effort estimation process.

This method needs to be applied in real software development companies in Chile, and its performance needs to be formally measured. For the moment, it represents a general guide for the expert estimator for lowering risks and support estimations. CWADEE is a repeatable and adjustable software engineering method, so it has the potential of being more and more accurate as we apply it once and again. Currently, we are applying CWADEE in small size Web-based information systems in Chile; however, we could extend it to different scenarios in the future.

Acknowledgements

This work has been partially funded by Fondo Nacional de Ciencia y Tecnología de Chile (FONDECYT) grant No. 103-0959. We acknowledge J.C.Trujillo of Motorola Chilean Center for Software Technology for his contributions.

References

[1] B. Boehm, "Software Engineering Economics", Prentice-Hall, January 1982.

[2] B. Boehm, "Anchoring the Software Process", IEEE Software, Vol. 13, No. 4, pages 73 -82, July 1996.

[3] B. Boehm, E. Horowitz, R. Madachy, D. Reifer, B. K. Clark, B. Steece, A. Winsor Brown, S. Chulani and C. Abts, "Software Cost Estimation in COCOMO II", Prentice-Hall, 1st edition, January 2000.

[4] B. Boehm and R. Fairly, "Software Estimation Perspectives". IEEE Software, Vol. 17, No. 6, pages. 22 - 26, Nov./Dec. 2000.

[5] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad and M. Stal, "Pattern-Oriented Software Architecture, Volume 1: A System of Patterns", John Wiley & Son Ltd; 1st edition, August 1996.

[6] A. J. C. Cowderoy, "Size and Quality Measures for Multimedia and Web-Site Production". Proc. of the 14th International Forum on COCOMO and Software Cost Modeling, Los Angeles, CA, October 1999.

[7] M. Fowler, "Patterns of Analysis: Reusable Object Models". Object-Oriented Software Engineering Series. Addison-Wesley. 1996.

[8] M. H. Halstead, "Elements of Software Science", Elsevier North Holland, Dordrecht, The Netherlands, 1977.

[9] R. Hay and R. Barker, "Data Model Patterns: Conventions of Thought". Dorset House Publishing. 1996.

[10] International Function Point Users Group, "Function Point Counting Practices Manual". Release 4.0, URL: http://www.ifpug.org/publications/manual.htm, 1994.

[11] G. Jiménez, G. Gioia, J. Pérez Herrera, R. Bertone, H. Ramón. "Estimating the Effort based on the Data Model" (in Spanish). IX Chilean Computer Science Workshop, Punta Arenas, Chile, November 2001.

[12] D. Lowe, "Web Engineering or Web Gardening?" WebNet Journal. Vol. 1, N° 1 January - March 1999.

[13] J. Matson, B. Barret and J. Mellichamp "Software Development Cost Estimation Using Function Points". IEEE Transactions on Software Engineering, Vol. 20, No. 4, pages 275-287, April 1994.

[14] G. Parra. "Estimating the Development Effort and Duration of Web Projects". Computing Engineering Thesis (in Spanish). CS Department, University of Chile, January 2003.

[15] D. Phillips, "The Software Project Manager's Handbook". IEEE Computer Society Press, 1998.

[16] D.J. Reifer, "Web Development: Estimating Quick–to-Market Software". IEEE Software, Vol. 17, No. 6, pages 57 - 64, November. December 2000.

[17] G. Rule, "Bees and the Art of Estimating". IEEE Software, Vol. 17, No. 6, page 2, November 2000.

[18] E. Sacre, "A Methodology To Develop Web Applications in Small and Medium Size Enterprises" (in Spanish). Master Thesis. CS Department, University of Chile, July 2003.

[19] W. Stein, "A Web Software Process for Small or Medium-Sized Projects, Focused on the Chilean Scenario". Engineering Thesis (In Spanish). CS Department, University of Chile, April 2003.

[20] J.C. Trujillo, "Generating Information Systems User Interfaces Using Design Patterns". Master Thesis (in Spanish). Computer Science Department. Pontificia Universidad Católica de Chile, 2000.