Software Qual J (2006) 14:339–357
DOI 10.1007/s11219-006-0039-0

Enhancing input value selection in parametric software cost estimation models through second level cost drivers

Juan J. Cuadrado-Gallego · Luis Fernández-Sanz · Miguel-Ángel Sicilia

Abstract Parametric cost estimation models are widely used effort prediction tools for software development projects. These models are based on mathematical models that use as inputs specific values for relevant cost drivers. The selection of these inputs is, in many cases, driven by public prescriptive rules that determine the selection of the values. Nonetheless, such selection may in some cases be restrictive and somewhat contradictory with empirical evidence, in other cases the selection procedure is somewhat subject to ambiguity. This paper presents an approach to improve the quality of the selection of adequate cost driver values in parametric models through a process of adjustment to bodies of empirical evidence. The approach has two essential elements. Firstly, it proceeds by analyzing the diverse factors potentially affecting the values a cost driver input might adopt for a given project. And secondly, an aggregation mechanism device for the selection of input variables based on existing data is explicitly devised. This paper describes the rationale for the overall approach and provides evidence of its appropriateness through a concrete empirical study that analyses the COCOMO II DOCU cost driver.

Keywords Parametric estimation models · Cost drivers · Software projects · COCOMO II · Empirical adjustment

J. J. Cuadrado-Gallego · M.-Á. Sicilia
E.U. Politécnica, Campus Universitario, Carretera Madrid-Barcelona, Km. 33,600, C.P. 28871, Alcalá de Henares (Madrid)
e-mail: jjcg@uah.es

M.-Á. Sicilia
e-mail: msicilia@uah.es

L. Fernández-Sanz (✉)
Dep. Sistemas Informaticos, Universidad Europea de Madrid, C/Tajo s/n, 28670 Villaviciosa de Odon, Madrid, Spain
e-mail: luis.fernandez@uem.es

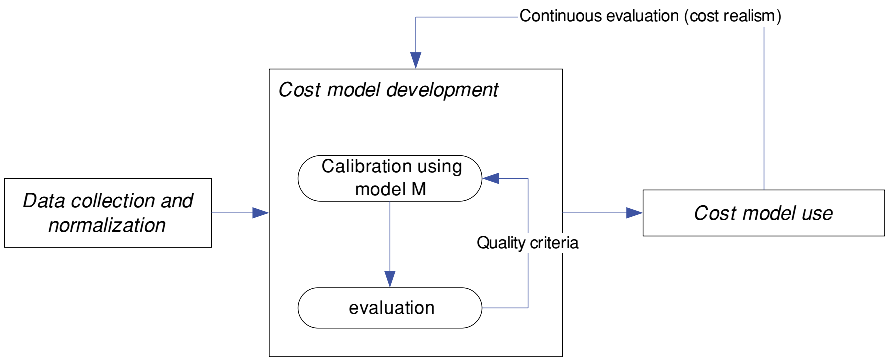

Fig. 1 Main steps in the development of parametric CERs

1 Introduction

Parametric software cost estimation models are one of the principal effort prediction methods used in software project management (Boehm, Abts and Chulani, 2000). Parametric estimation is based on the historical use of project databases and expert knowledge to obtain a *Cost Estimation Relationship* (CER), i.e. a mathematical model that derives effort estimates from selected project attributes. CERs use as input a set of variables (cost drivers) that are known to significantly affect the overall effort required. According to the *Parametric Estimating Handbook* (PEI, 1999), the main elements of the development of a parametric model entails a first phase of data collection and normalization followed by the development of the cost model. The latter includes calibration and validation of the model. Figure 1 depicts these main phases.

When using this kind of cost estimation models, we can consider three major aspects as determinants of the quality of the resulting models, besides of the quality and meaningfulness of the available data:

(a) The CER expression used, not only regarding the mathematical model *M* selected but also regarding the collection of input variables selected. A considerable amount of research on parametric estimation models has focused on this aspect (e.g. Farr and Zagorski, 1965; Herd et al., 1977; Putnam, 1978; Baylei and Basili, 1981; Boehm, 1981; Jensen, 1983; Rubin, 1983; Putnam and Mayers, 1992; Boehm et al., 1995).
(b) How models are calibrated. There are also many calibration studies that introduce different techniques for better adjustment quality and the management of uncertainty (e.g., Shrum, 1997; Fischman, 1997; Chulani et al., 1998, 1999a,b; Ferens and Christensen, 1999; Mertes et al., 1999; Sicilia et al., 2005).
(c) The methods used to select the values for the cost drivers for each concrete project. Surprisingly, this aspect has attracted less attention. In this direction, Baik and Boehm (2000) described the decomposition of the COCOMO II TOOL cost driver, which was later applied as a technique to improve predictive accuracy (Baik et al., 2002). Similarly, Cuadrado et al. (2000) described the analytical decomposition of the COCOMO II DOCU variable. Other texts as (DoD, 1999) show that the correct assessment of the cost drivers inputs for each particular project is acknowledged as a important milestone in parametric estimation.

When refering to aspect (c), Cuadrado et al. (2005) demonstrated that the impact of this factor may be larger than previously mentioned factors in a significant amount, and that this may be even increased thorough the project development.

This paper describes a novel approach based on the concept of "second level" cost drivers to improve the quality of the selection of the cost driver values and the results of effort estimations The idea of multilevel process has been used before in other areas of software process and software metrics, e.g. numerous examples similar to the FCM (factor criteria metric) model for software quality evaluation (McCall et al., 1977). Other examples of the application of multilevel process could be found in (Prather, 1995).

The rationale for this approach is that second-level or subordinate aspects of a cost driver can be used for its indirect assessment, provided that empirical data for them is available. In other words, the approach departs from the analysis of the diverse factors potentially affecting the values a given cost driver might take for a given project. Then, an aggregation device for the selection of input variables is explicitly devised, as a result of a process of adjustment to empirical evidence. This technique is thus useful both for the empirical validation of existing cost driver input selection procedures and also as a technique to provide an alternative to them. An specific empirical study regarding the COCOMO II DOCU variable that considers the impact of developing software documentation (Boehm et al., 2000) is described as an illustration of the appropriateness of the approach to obtain an enhanced input selection method.

The rest of this paper is structured as follows. Section 2 describes the details of the method of empirical assessment just mentioned, providing the details of its application in generic terms. Then, the method is evaluated in a concrete setting in which an available database is used for the assessment of the DOCU cost driver, resulting in rating levels that are different to those provided by the COCOMO-II model. Conclusions and future research directions are provided in Section 4. Finally, the details of the computation of the rating for the case study are provided as an appendix to the paper along with complimentary information about different aspects related to the example and mathematical foundation.

2 A second level cost driver-based assessment technique

In this section, a method to improve the cost drivers rating level selection is described. The first subsection provides the rationale for the method. Then, the second subsection details the steps required for the application of the technique. As a definition, a cost driver directly included in the CER is designated by the expression First Level Cost Driver (FLCD).

2.1 Rationale for the use of second level cost drivers

The selection ratings of some cost drivers used for a parametric estimation model may depend on the value of a set of factors, which affect such variables (Cuadrado et al., 2000; Baik et al., 2000, 2002; Sicilia et al., 2005). For example, the product complexity is a widely used FLCD in different parametric cost estimation models. Its rating level for an specific project depends specifically in the COCOMO II Post-Architecture Model on very different and heterogeneous factors with their own weights in the rating level selection, namely: control operations, computational operations, device dependent operations, data management, operations user interface and management operations. These ones can be considered as factors that contribute or are included to some extent in the FLCD, which is of a more abstract nature. Another example of cost driver that is usually considered through constituent aspects is required *usability*, usually broken down in sub-aspects as efficiency, effectiveness and user satisfaction (van Welie et al., 1999), and for which Crespo et al. (2004) devised a fuzzy aggregation scheme in a particular application.

The factors that are candidate to decomposition in more concrete aspects or *Second Level Cost Drivers* (SLCD) have two main properties:

- They encompass heterogeneous facets, i.e., they reflect or summarize the effect of different more concrete characteristics. Therefore, for a specific project, each of these characteristics will eventually have its own assessment, being possibly independent (i.e. no correlated) with the other factors affect the same FLCD.
- Each of the second-level characteristics or factors will have its own relative influence (weight) on the rating selected for the FLCD in question and, therefore, on the final estimated values for the project being estimated.

Neglecting to inquiry about SLCD would imply an oversimplification in many cases. This may lead to erroneous or biased selection of rating for the current project, so the estimations obtained would not be adequate. As a consequence, we can conclude that these secondary variables have a potential significant effect on the final estimation, as they determine to a certain extent the value of its associated FLCD. Even though they are not directly included into the CER, they can be referred to as (second level) cost drivers because they are significant for estimations and appear as independent inputs when considered in the parametric model.

For example, let us consider a generic lineal parametric model:

$$y = a_0 + a_1 x_1 + a_2 x_2$$

where y is any dependent variable as effort, and a_0, a_1, a_2 are the model coefficients affecting linearly two indepedent cost drivers x_1 and x_2. They can be consider as FLCD because they are directly included in the equation. We could have the following three rating levels obtained from a calibration process with their corresponding numerical values for both of them (see Table 1),

Let us suppose now that the rating level for x_1 in each specific project will depend only of one factor that can be obtained by applying the rules in Table 2.

Continuing the example, let us state that the rating level for in each specific project will depend on three hetereogeous factors, that are assessed using other specific rules. As these new factors do not appear directly in the CER and are associated and determine the value for, they can be considered as SLCD. In the case that corresponds

Table 1 Rating level numerical values for x_1 and x_2 cost drivers

Rating	Low	Nominal	High
x_1 Numerical value	1.15	2.56	3.78
x_2 Numerical value	0.54	1.05	4.56

Table 2 Rating level selection criteria for x_1 cost driver

Rating	Low	Nominal	High
Selection criteria	Some of the lifecycle needs are not covered	Correct for the lifecycle needs	Excessive for the lifecycle needs

to usability, we could consider for example three interdependent aspects: efficiency (z_1), learnability (z_2), and satisfaction (z_3). In consequence, estimation will typically include an aggregation stage in which partial estimations of importance regarding different attributes would need to be summarized in an overall value of usability. Crespo et al. (2004) have described an example of using an OWA operator (Yager, 1988) to model such aggregation for a given project. Nonetheless, the procedure described in that paper is highly dependent on the nature of usability evaluation, since it applies the commonly used Nielsen model for predicting the effort in terms of the number of users required in usability testing (Nielsen, 1999). This raises the need for a general-purpose technique that may be tailored to each specific case, but retaining a systematic structure that justifies the decisions adopted. Such a method is described below.

2.2 Second level cost drivers application process

The mathematical method of rating selection for cost drivers based on SLCD is made up of generic and a specific process, carried out in sequence.

2.2.1 Generic process

This process consists in three sequential steps, that must be carried out once for the parametric cost estimation model selected:

Step 1. Study if each FLCD in the model CER could be better assessed in terms of SLCDs, and, in the positive case, define them in a formal way. For example, "analyst experience", a commonly used first level cost driver, can be determined univocally by only one magnitude: the average experience of the analysts' team measured, instance, e.g., in months. But other also commonly used first level cost driver like "documentation developed through the project", could be considered as dependent on some second level cost drivers (details of the behavior of this cost driver will be described below), as documentation size, documentation complexity and others. The documentation input value for a specific project should be assessed by aggregating the values of its associated SLCDs.

Step 2. Build a qualitative and quantitative rating scale for each SLCD (obtained in step one) that allows us to determine its value for a specific project. For example, for each of the documentation cost drivers mentioned above, a description and a qualitative rating scale should be defined. "Documentation size" could be described as "the amount of documentation developed during the project development regarding the software process covered by project documentation" and its ratings could be the ones showed in Table 3. The numerical values have taken into account the mathematical aggregation operator used (which will be detailed below).

Table 3 Rating level selection criteria and numerical values for SLCD documentation size

Rating	Numerical value	Description
Very low	0	Only the basic development documentation: user required documents, software requisites, code documents and user manual
Low	2	More refined technical documentation which includes functional analysis and low rating design
Nominal	4	In addition to the previous documents, documentation related to software project management: description of the project plan, estimation documentation, follow-up reports and final analysis
High	6	In addition to the previous documents, documentation related to quality assurance of the plans and tests
Very high	8	In addition to the previous documents, documentation related to auditing, management plans and other documentation

Table 4 Rating level selection criteria and numerical values for FLCD documentation

Rating	Aggregated numerical value of the SLCD	Description
Low	(0,8)	Some of the lifecycle needs are not covered
Nominal	(9,15)	Correct for the lifecycle needs
High	(16,24)	Excessive for the lifecycle needs

Step 3. Build a scale of numerical values for each FLCD that has associated SLCDs, called a Rating Level Selection Scale. Each one of the numerical values into the scale (Rating Level Selectors) will be associated with a specific rating level of the FLCD. This will enable the selection of the appropriate FLCD rating level for a specific project. This will be done starting with the values which, for this project, have SLCD associated with the FLCD studied and using a mathematical aggregation operator obtain the relevant rating level selector. For example, if we consider documentation (provided that Table 2 rating levels are used for selection) a new column containing the rating level selection scale could be added as showed in Table 4.

2.2.2 Specific process

It consists of three sequential steps that should be done for each project estimated with the model, i.e. it is the procedure actually used in practical estimation settings. In the next section this process will be detailed through the case study.

Step 1. Select the rating of each SLCD associated with the studied FLCD and determine the corresponding numerical value for the project.
Step 2. Using the SLCD values obtained in step one, calculate the numerical value of the rating selector for the FLCD using the devised mathematical aggregation operator.
Step 3. Using the value obtained in step two, select the FLCD rating level.

Figure 2 depicts the main differences between the common technique that uses only first-level input values (FLIV) for cost drivers (right part of the figure) and the technique described here that uses second-level input values to derive them (left part

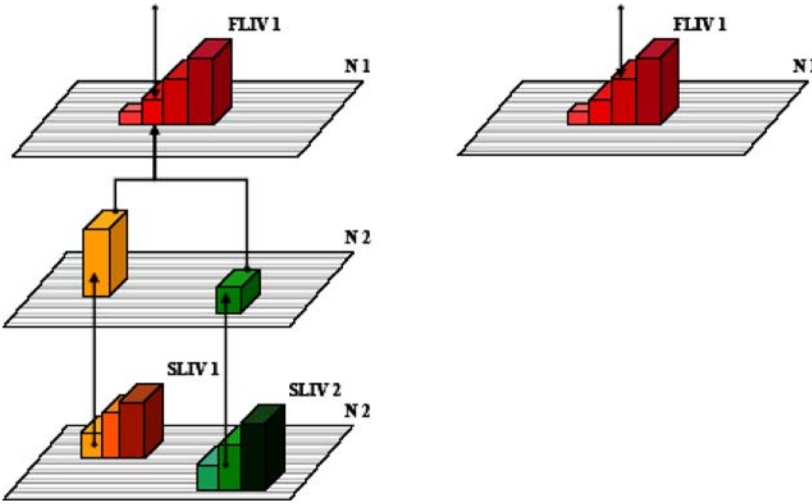

Fig. 2 Graphical description of the differences between the single-level and the second-level cost driver approach

of the figure). In the second-level approach, the values selected for the first-level cost driver (N1) are obtained following the steps of selection and aggregation of input values at the second-level (N2).

So the main problem that should be addressed to derive the rating selection procedure is the need of devising an aggregation operator and carefully selecting the rating values for SLCD. This should be accomplished by adjusting them to empirical evidence whenever project data is available in a process of local calibration for each cost driver.

3 Evaluation of the cost driver value selection procedure

An experiment using the COCOMO II DOCU cost driver is described in this section as an illustration of the method described in the previous one also as supporting evidence of the appropriateness of the decomposition approach. The following six steps were carried out:

Step 1 The COCOMO II Post-Architecture parametric model (Boehm et al., 2000) was considered for the experiment due to its public nature and its wide acceptance, more concretely the "documentation generated during the lifecycle of the software" (DOCU) FLCD was selected as a case study. The COCOMO II model allows us to estimate the effort and schedule involved in a software development project. It is an update of the COCOMO series after Ada-COCOMO and the original COCOMO (also know as COOMO 81). The latest COCOMO II is composed of three sub-models named Application Composition, Early Design and Post-architecture that will be used at different stages of the development process as more information becomes available. The Post-Architecture model, when the architecture of the system is available, is the one that has been studied and developed in more detail.

This model is based on the following equation:

$$PM = a \cdot \text{size}^{b+0.01\cdot\sum_{i=1}^{5} SF_i} \cdot \prod_{i=1}^{17} EM_i$$

where *PM* is the development effort in Person-Months, and *a* and *b* are constants which values are 2.94 and 0.91 respectively in the calibration provided by the authors of the method. The variable Size is the size of the system in KLOC (thousand lines of code), and the EM_i variables represent cost drivers which are defined by a set of rating levels with their corresponding set of numeric values. Finally, the equation's SF_i are called "scale factors" and also present a rating level with their associated numerical values.

The first level variable "*Documentation Match to Lyfe-Cycle Needs*" (DOCU) was selected as a case study since empirical data was available that allowed the separation of their influence from the overall effort estimations. In addition, this variable is used in many estimation models apart from COCOMO. COCOMO-II provides a rating scale in which the values are selected according to the suitability of the project documentation to the needs of the software process that it is followed. The rationale behind is tha fact thatreducing the effort spent in documentation (e.g. in an attempt to reduce costs) may lead to increased costs in maintenance.

Step 2 A set of 17 business applications with similar characteristics were developed, with a previous statement of their characteristics, as described in the method above (It is just a coincidence that the number of developed applications is the same as the number of cost drivers in COCOMO model). The standard used in the development of the documentation was in all cases the one of the European Space Agency (ESA, 1991), which recommends the generation of the documents listed in Table 5.

Table 5 ESA recommended documents

Acronym	Document title
SQAP	Software Quality Assurance Plan
SCMP	Software Configuration Management Plan
SPMP	Software Project Management Plan
URD	User Requirements Document
SVVP	Software Verification and Validation Plan
SRD	Software Requirements Document
PFI	Initial Function Points Document
UID	User Interface Document
ADD	Architecture Design Document
DDD	Detailed Design Document
PFF	Final Function Points Document
SVR	Software Verification Report
DDD	Detailed Design Document
STD	Software Transference Document
SUM	Software User Manual
AUD	Audit Document
PHD	Project History Document

Table 6 Rating level numerical values and selection criteria for FLCD documentation

Rating	Numerical value	Description
Very low	0.81	Many of the lifecycle needs are not covered
Low	0.91	Some of the lifecycle needs are not covered
Nominal	1.00	Correct for the lifecycle needs
High	1.11	Excessive for the lifecycle needs
Very high	1.23	Very excessive for the lifecycle needs

Step 3. The documentation artifacts generated and the actual effort spent in the projects were recorded and analyzed. This allowed the determination of a numerical value for the input factor DOCU for each project. For all the projects the time spent in the documentation development was recorded. This was done in order to calculate the effort percentage of effort dedicated to develop the project documentation when compared to the total project development effort. The final results pointed out that the effort percentage dedicated to the documentation of the different projects from the total project development effort ranged from a minimum value of 12.34% and a maximum value of 31.67%, with an average value of 24.47%.

Using the computation procedures described in detail in Appendix A, the adjustment value for the COCOMO II Post-Architecture variable DOCU was derived. The value obtained from the experiment for DOCU ranges between 1.46 for the higher value and 1.13 for the lower, with a most probable value of 1.32. This factor was adjusted according to previously published studies (NASA 1990, 1995, 1996). These studies indicate that the average effort needed to develop the documentation of a project is 11% of the total. Taking these results into account and adjusting the actual values obtained for the coefficient DOCU, the determined value was obtained is 1.01 for the lower and 1.26 for the higher, with 1.15 as the most likely value.

Step 4. The COCOMO II Post-Architecture model was used to derive effort estimates for the projects, using the original rating levels and the rating selection criteria for the FLIV DOCU. The COCOMO 2000.0 calibration provides the numerical values described in Table 6.

The documents created for the projects developed for this study should be classified as “Very High” according to the criteria in Table 6 or, using a broad margin of error, as “High”, considering that all of them were experimental. In other words, they were carried out without a rigorous following of the selected ESA software process and without any provision for maintenance. Nonetheless, the standard for the documentation used was conceived for projects with diametrically opposing characteristics such as high ratings of quality and reliability, exhaustive tests and continuous maintenance. In this case, if the ratings in Table 6 are used, the numerical values corresponding to the COCOMO II 2000.0 calibration of the model proposes for these ratings are 1.23 for “Very High” and 1.11 for “High”. Comparing these values with those obtained in our case study, it can be concluded that they are very similar. The divergence appears in the lower limit experimentally obtained for the variable DOCU, since the numerical value 1.01 corresponds to the medium rating of the COCOMO II model. If we consider the project characteristics and the documentation rating selection criteria for the COCOMO II Post-Architecture, it would be impossible to select this rating for any of the projects presented. Summarizing

Table 7 Rating level numerical values and selection criteria for SLCD S_DOCU

Rating	Numerical value	Description
Very low	0	Only draft no planed documents
Low	2	Software requisites, code documents, user manual
Nominal	4	In addition to the previous document, functional analysis and design documents.
High	6	In addition to the previous documents, documentation related to software project management: description of the project plan, estimation documentation, tracking reports
Very high	8	In addition to the previous documents, documentation related to quality assurance

the above, a value lower than 1.11 could never be obtained. This is a case in which existing rating levels diverge from a body of empirical evidence.

Step 5. The rating selection criteria determined by the consideration of the SLCDs are used to produce alternative estimates. The determination of the numerical value for the input variable DOCU obtained from this method was the result of application of the above described general and specific processes:

- Firstly, the SLCDs associated to FLCD DOCU were analyzed. The structured expert consensus method Delphi (Linstone and Turoff, 2002) was used to determine the three fundamental factors which influence the effort used in the documentation development:
  1. The *size* of documentation. SLCD S_DOCU
  2. The *complexity* of the documentation. SLCD C_DOCU
  3. The use of documentation *standards* and *traceability.* SLCD ST_DOCU

  To select each one of these SLCD, a set of ten academic and industry experts in software engineering were consulted with a two round Delphi approach. In the first round they were asked to define which are the main factors that could affect the FLCD DOCU following their expert criteria. In the second round, they reached an agreement in the three selected ones. As an example one of the proposed but not selected SLDC was the type of documents.
- Then, the characterization of the ratings for each of them was determined. Tables 7 to 9 providethe results of this process.

After that initial characterization, the variables proposed by COCOMO II Post — Architecture (Table 6), were substituted with quantitative criteria. Concretely, the numerical values ratings provided in Table 10 were used.

Once the three generic steps were completed, the specific process was carried out for the rating selection for each SLCD associated with the FLCD DOCU:

SLCD S_DOCU: Based on the sandard composition that followed, the Very High rating level, which associated numerical value is 8 (Table 6), was chosen for this first SLIV.

SLCD C_DOCU: For this second SLIV, the rating chosen ranged “Low”, and in certain cases, “Nominal”. This was because, in some cases, the documentation was

Table 8 Rating level numerical values and selection criteria for SLCD C_DOCU

Rating	Numerical value	Description
Very low	0	Plain text done without tools
Low	2	Plain text generated with common text editors
Nominal	4	Plain text and diagrams generated with CASE tools
High	6	In addition to the previous documents, multimedia documents generated authoring tools
Very high	8	In addition to the previous documents, documents with new models and methods written specifically for this

Table 9 Rating level numerical values and selection criteria for SLCD ST_DOCU

Rating	Numerical value	Description
Very low	0	Standards are not used to prepare the documentation of the software project. The documents are not related.
Low	2	Only some of the documents on technical development adapt to a standard. The documents are not related
Nominal	4	All the documents on technical development adapt to a standard. The documents are related
High	6	All the documents on technical development and management adapt to a standard. Both kind of documents are not related
Very high	8	All the documents on technical development and management adapt to a standard. Both kinds of documents are related

Table 10 Rating level numerical values and selection criteria for SLCD Use of standards and traceability of documentation

Rating	Numerical value
Very low	0–4
Low	5–9
Nominal	10–14
High	15–19
Very high	20–24

Table 11 Rating level selected for SLCD

SLCD	Rating	Numerical value
SLCD S_DOCU	Very high	8
SLCD C_DOCU	Low, nominal	2, 4
SLCD ST_DOCU	Very high, high, nominal	8, 6, 4

prepared using only a text editor but, normally, CASE tools were used. Therefore, the associated numerical values for this variable, using Table 7, are 2 or 4.

SLCD ST_DOCU: If we again consider the composition of the ESA standard for this SLIV, we are forced to choose the "Very High" rating, (with associated numerical value 8). However, as the projects were experimental, some of them did not develop some of the documents (among these, SQAP was always underdeveloped). Therefore, the "High" or "Nominal", with a numerical value of 6 or 4 (Table 8), could also be selected for this variable for specific projects.

The selected rating levels for SLCD are presented in Table 11.

After selecting the rating of each SLCD, the numerical value of the rating selector for the FLCD DOCU was computed.

The set BDOCU (see Appendix B) obtained from the previous step is:

$$\begin{aligned} B_{\text{DOCU}} &= \{\text{S} - \text{DOCU}, \text{C} - \text{DOCU}, \text{ST} - \text{DOCU}\} \\ &= \{b_{1_{\text{DOCU}}}, b_{2_{\text{DOCU}}}, b_{3_{\text{DOCU}}}\} = \{Z_{1_{\text{DOCU}}}, Z_{2_{\text{DOCU}}}, Z_{3_{\text{DOCU}}}\} \\ &= \{(8), (2, 4), (4, 6, 8)\} \end{aligned}$$

Then, the equation provided in Appendix B for the aggregation of second level values into first level ones is used

$$y_{k_{a_i}} = \sum_{l=1}^{r} w_l (z_{m_l})^{f_l}$$

with $w_l = 1$ and $f_l = 1 \forall l$ (the reason to adopt these values is that the equation above is a generic equation that has been devised with the purpose of introducing different weight for each one of the SLCD in the process to select the rating level for the FLCD associate (this can be appreciated in Figure 2). In this case we have establishes that all the SLCD have the same weight so the values for w and f adopt the value 1 and introduce the values z_{m_l} from the set B_{DOCU} for this experiment in particular. From this we get the following set of rating selectors Y_{DOCU} (see Appendix B)

$$Y_{\text{DOCU}} = \{14, 16, 18, 20\}$$

Using Table 10, the *Medium, High* and *Very High* ratings for the FLCD DOCU ratings are finally computed.

If Table 4 is used, we get the set X_{DOCU} (see Appendix B)

$$X_{\text{DOCU}} = \{0.81, 0.91, 1.00, 1.11, 1.23\}$$

and consequently the correspondence C_{DOCU} with the sets Y_{DOCU} and X_{DOCU} is the following one:

$$C_{\text{DOCU}} = \{(1.00, 14), (1.11, 16), (1.11, 18), (1.23, 20)\}$$

This is determined by establishing a relationship between Tables 6 and 10. This means that, using the mathematical model described in the second-level technique, the values chosen for the FLIV DOCU for the projects would be 1.23 "Very *High*", 1.11 "High" and 1.0 "Nominal".

Step 6. The values obtained in steps 4 and 5 for the FLIV DOCU are compared with those ones obtained in step 3 for the ratings corresponding to each project. The degree of coincidence is determined and, in consequence, the accuracy of each of the rating selection methods is compared. For the experiment the results of the technique proposed here (based on the use of the SLCD) are more coherent with the experimental evidence than the other ones. This can be explained by the use of the original COCOMO ratings tend to preclude the selection of the 1.01 rating, a level that, as we have explained above, is required due to the characteristics of performance of some projects. This fact is important for managers that need to do estimations for such projects and with the traditional method, the selection of the "Nominal" rating level could be impossible even in the case that, taking into account his experience and the knowledge of the specific projects to be estimated with COCOMO, indicates him that Nominal is the adequate rating level. The use of the SLCD model allows him to balance the effect of the different SLCD in order to obtain the Nominal value for the FLCD DOCU.

4 Conclusions

The above described technique for modeling cost driver input value selection allows the linking of the parametric model to a body of empirical evidence that, in turn, can be used to formulate some overall criteria about the quality of input assessment in parametric estimation. In an specific way, the quality of an input selection procedure or technique can be assessed in terms of the following aspects:

- Its *clearness* and lack of ambiguity, if a non-automated procedure is provided. The use of SLCD avoid any ambiguity included in the FLCD definitions. SLCD definitions should be more concise and precise. This is based on the fact that each different aspect of a FLCD that could introduce ambiguity in the definition of the criteria description for a rating level is considered as a different SCLD with its own definition. For example,for the FLCD used in the experiment, DOCU, one manager could determine himself if the quantity of the documentation is correct o excessive for the lifecycle needs (Table 6); two different managers could have two different opinions for the same quantity of documentation for the same project. However, following with the same example, it is easy to verify by reading of the description of the rating level selection criteria that its more difficult to doubt which of the three cost drivers presented in Tables 7, 8 and 9 is appropriate.
- Its *consistency* with empirical evidence. In the experiment we have observed that the use of SCLD allow us the determination of rating levels for the cost drivers with more consistency with the values actually observed. This is the case of the "Nominal" rating level for some projects.
- Its capability to *cover* the range of situations that actually take place in specific project settings. The combination of the values of different SLCD in the aggregation equation give us the possibility of determining the value of the FLCD ratting levels (almost impossible of being obtained with the traditional selection method). A clear example in the experiment where the rating level "Nominal" that the second level selection method allow to obtain for some of the projects studied: this selection was impossible to be done with the traditional method.

Clearness, consistency and coverage address different aspects of the notion of quality in cost driver input selection, and could be used as general criteria for auditing or evaluation of parametric estimation models. Clearness is only meaningful when human-oriented procedures are provided for selection. The most typical case is the one of COCOMO, where input selection for many cost drivers relies on judgment about some categories.

Effort required to obtain the effort estimation using the SLCD rating level estimation method is not as long as with the use of the traditional method. If we consider again the example used in the experiment, the additional effort to read three tables instead of one its no too much compared with the advantages in the precision obtained in the equations and makes it worthless of extra effort. There are only additional work in the definition of the SLCD for each FLCD in each model, but this work its done only once for each FCLD in a model.

Appendix A. Calculation of the total effort adjustment factor of the project attributed to the development of documentation

We will assume a non-lineal parametric equation such as the one used in the COCOMO II Post – Architecture model.

$$E_N = A \times (s)^B \tag{A.1}$$

where E_N is the average effort in man-months, s is the size of the product in thousands of lines of source code, and A and B are two adjustment factors.

If an adjustment factor d is introduced, which represents the contribution of the software documentation to the overall effort spent, the equation will be changed as expressed in (A.2):

$$E_D = A \times (s)^B \times d \tag{A.2}$$

where E_D is the effort developed in man-months, considering the effort dedicated to develop the software documentation.

In our experiment, we obtained the values for each project, E_D and E_d, the latter is the effort devoted to develop the documentation of the project and expressed as in (A.3), where p is the percentage of the total effort dedicated to software documentation.

$$E_d = E_D \times p \tag{A.3}$$

Therefore, the development effort expression that takes into account the consideration of documentation is (A.4).

$$E_D = \left(\frac{1}{1-p}\right) \times E_N \tag{A.4}$$

If the real numerical value p is known, we can therefore calculate the coefficient d, which should have been used on each product, from Equation (A.5).

$$d = \left(\frac{1}{1-p}\right) \tag{A.5}$$

Appendix B. Mathematical foundations of the method

A is the non null set its n elements are the FLCD of the model

$$A = \left\{ \begin{array}{l} \{a_i | i \in \aleph \wedge (1 \leq i \leq n) \\ n \text{ depending on the model} \end{array} \right\} \tag{B.1}$$

This should be non-empty set since, the initial established assumption is that the method will be applied to parametric models. This imply the use of a not null set of input variables. The number of variables used would vary according to the model.

a_i is the FLCD whose value rating is the finite not null set X_{a_i}

$$Xa_i = \left\{ \begin{array}{l} x_{j_{a_i}} | x_{j_{a_i}} \in \Re \wedge j \in \aleph \wedge (\{1 \leq j \leq p) \\ p \text{ depending on the FLCD } a_i \text{ and the model} \end{array} \right\} \tag{B.2}$$

Each of the FLCD $a_{i,}$ (which value depends on the model and the calibration chosen) will include a set of corresponding real numerical values. These values are classified as a set of ratings, which also depends on the model used.

Y_{a_i} is the finite not null set whose elements q are the rating selectors of the FLCD a_i of the model.

$$Y_{a_i} = \left\{ \begin{array}{l} y_{k_{a_i}} | y_{k_{a_i}} \in \Re \wedge k \in \aleph \wedge (1 \leq k \leq q) \\ q \text{ depending on the FLIV } a_i \end{array} \right\} \tag{B.3}$$

The rating selectors corresponding to a specific FLCD only depend on the studied variable, as they will determine the associated numerical values correspond to each FLCD described in the set B.2 through the correspondence set (Equation B.4) described in the paragraph below.

C_{a_i} is the not null product set $X_{a_i} \times Y_{a_i}$, which defines the correspondence between $X_{a_i} e Y_{a_i}$

$$C_{a_i} = \{(x_{j_{a_i}}, y_{k_{a_i}}) | \forall x_{j_{a_i}} \in X_{a_i} \wedge y_{k_{a_i}} \in Y_{a_i}\} \tag{B.4}$$

B_{a_i} is the set whose elements are the SLCD of the FLCD a_i

$$B_{a_i} = \left\{ \begin{array}{l} b_{l_{a_i}} | l \in \aleph \wedge (1 \leq l \leq r) \\ r \text{ depending on the FLCD } a_i \end{array} \right\} \tag{B.5}$$

This set can be empty in some cases since, for a specific parametric model, it may have FLCD for which it is not necessary to define a set of second level variables. Its rating will depend on one unique aspect which is adequately reflected in the associated selection rating criteria. Therefore, it does not depend on the set of heterogeneous characteristics or second level variables. As a result, the number of second level variables, which will influence the value of a first level variable for a specific project, depends only on the variable considered.

$b_{l_{a_i}}$ is the SLCD whose rating values is the not null finite set $Z_{l_{a_i}}$

$$Z_{l_{a_i}} = \left\{ \begin{array}{l} z_{m_{l_{a_i}}} | z_{m_{l_{a_i}}} \in \Re \wedge m \in \aleph \wedge (1 \leq m \leq s) \\ s \text{ depending on the SLCD } b_{l_{a_i}} \end{array} \right\} \quad \text{(B.6)}$$

Each second level variable has a set on ratings whose number depends only on the variable considered.

Therefore, if we apply the non- lineal equation

$$y_{k_{a_i}} = \sum_{l=1}^{r} w_l (z_{m_l})^{f_l} \quad \text{(B.7)}$$

we can obtain the ratings selector value y_k for the FLCD a_i and, consequently, through set C_{a_i}, its real value $x_{j_{a_i}}$. w_l is the weight factor associated with the SLCD b_l and shows the relative importance of each of its SLCD with which it obtained the specific FLCD rating. f_l is the exponent associated with the SLCD l.

Appendix C. Brief description of COCOMO II Post-Architecture model

The COCOMO II Post–Architecture model is based on a no lineal Equation (C.1) that obtains an estimation of the effort to be spent in the development of a software project.

$$E = A.(s)^{B+C\sum_{i=1}^{5} y_i}.\left(\prod_{j=1}^{17} x_j\right) \quad \text{(C.1)}$$

E is the overall project effort measured in MM (Man-Month); s is the product size measured in KLOC (thousands of source code lines); A, B, C, are constants; x_j is the value of the cost driver j; and y_i is the value of the scale factor i.

Each one of the 17 cost drivers represent a different feature of the analyzed project. Five of them are related to product features: RELY (Required product reliability), DATA (Size of the product database), CPLX (Product Complexity), RUSE (Product reusability) and DOCU (Product documentation). Three of them are related to platform features: (TIME, execution time restrictions), (STOR, main store restrictions) and (PVOL, Platform volatility). Six of them are related with Personal features: ACAP (Analyst capability), PCAP (Programmers capability), AEXP (Analyst experience), PEXP (Platform experience), LEXP (Language experience) and PCON (Personnel continuity). And finally, three of them are related to project features: (TOOL (Software tools utilization), SITE (Development localization) and SCED (Time needed to develop the project)). There is also a set of five scale factors: PREC (Precedents), FLEX (Development flexibility), RESL (Architecture, Risk resolution), TEAM (Staff cohesion) and PMAT (Process Maturity)).

The process to use both, cost drivers and scale factors, for a specific project again requires two steps. The first one uses a rating level selection criteria, usually showed in a table (i.e. Table 2). The adequate rating level for each cost driver and scale factor in the current project is selected using the table. The second step requires that the numerical values associated to the rating selected for each cost driver should be identified, (usually using a different table but, in same cases, it could be the same one, e.g. Table 6.)

As an example, we may consider that the quantity of documentation developed is excessive for our current project according to the software lifecycle needs. If we use Table 2, we obtain a rating level High and the first step is completed. In step two we consult Table 6 and obtain a value of 1.11.

Once numerical values for the cost drivers and scale factor have been determined, they are introduced in the Equation C.1 to obtain the value for *E*, the effort estimated for the project. In the example, the multiplier for DOCU is $x_4 = 1.11$ to be multiplied by the other 16 cost drivers, by *A* and the value of the size powered to the numerical value obtained after solving the scale factors expression.

A more extended explanation of how COCOMO II Post-Architecture model works can be found in Boehm et al. (2000).

References

Baik, J., Boehm, B., Steece, B. 2000. The Effect of CASE Tools on Software Development Effort. Proceedings of the 15th International Forum on COCOMO and Software Cost Estimation Cost drivers Studies. Los Angeles, California (USA). 24–27/10.

Baik, J., Boehm, B., Steece, B. 2002. Disaggregating and calibrating the CASE tool variable in COCOMO II. IEEE Trans. Software Eng. **28**(11) pp. 1009–1022.

Baylei, J., Basili. V. 1981. A Meta-Model for Software Development Resource Expenditures. In Proceedings of the Fifth International Conference on Software Engineering, pp. 107–116.

Boehm, B.W. 1981. Software Engineering Economics. Prentice Hall.

Boehm, B., Abts, C., Chulani, S. 2000. Software Development Cost Estimation Approaches-A Survey. Center for Software Engineering, University of California, Technical Report USC-CSE-2000–505.

Boehm, B., Clark, B., Horowitz, E., Madachy, R., Selby, R., Westland, C. 1995. Cost Model for Future Software Life Cycle Processes: COCOMO 2.0. Annals of Software Engineering. Special Volume on Software Process and Product Measurement. J. Arthur, S. Henry and J. Baltzer (eds.). Amsterdam: AG Science Publishers, (1): 45–60.

Boehm, B., Abts, C., Winsor B., A., Chulani, S., Clark, B., Horowitz, E., Madachy, R., Reifer, D., Steece, B. 2000. Software Cost Estimation with Cocomo II. Prentice Hall.

Chulani, S., Clark, B., Boehm, B., Steece, B. 1998. Calibration Approach and Results of the COCOMO II Post–Architecture Model. In Proceedings of the 20th Annual Conference of the International Society of Parametric Analysts (ISPA) and the 8th Annual Conference of the Society of Cost Estimating and Analysis (SCEA).

Chulani, S., Boehm, B., Steece, B. 1999. From Multiple Regression to Bayesian Analysis for COCOMO II. In: Proceedings of the 21st Annual Conference of the International Society of Parametric Analysts (ISPA) and the 9th Annual Conference of the Society of Cost Estimating and Analysis (SCEA).

Chulani, S., Boehm, B., Steece, B. 1999. Bayesian analysis of empirical software engineering cost models. IEEE Transactions on Software Engineering, **25**(4):513–583.

Cuadrado-Gallego, J.J., Ernica E., Sánchez, M., Guzmán, J., Amescua, A. 2000. The cost estimator DOCU: An empirical and theoretical study. Proceedings of the 15th International Forum on COCOMO and Software Cost Estimation Cost Drivers Studies. Los Angeles, California (USA), pp. 24–27/10.

Cuadrado-Gallego, J.J., Marbán, O., Sánchez, M., Garcín, L. (2005) The importance of rating level selection for input variables to determine accurate estimations in parametric mathematical models. Journal of Cost Analysis and Management, Summer 2004, pp. 12–24.

Crespo, J., Sicilia, M.A., Cuadrado, J.J. 2004. On Aggregating Second-Level Software Estimation Cost Drivers: A Usability Cost Estimation Case Study. Proceedings of IPMU, pp. 1255–1260.

Department of Defense (DOD), United States 1999. Parametric estimating handbook, 2nd Edition.

ESA, 1991. European Space Agency. Software Engineering Standards, ESA PSS-05-0 Issue 2.

Farr, L., Zagorski, H. 1965. Quantitative Analysis of Programming Cost Factors: A Progress Report. In A. Frielink (ed.): Proceedings of the ICC Symposium on Economics of Automatic Data Processing. Amsterdam: North-Holland, Holland.

Ferens, D., Christensen, D. 1999. Calibrating software cost models to department of defense databases — a review of ten studies. Journal of Parametrics, **XIV**(1):33–52.

Fischman, L. 1997. Calibrating a Software Evaluation Model. In: Proceedings of the ARMS Conference.

Herd, J., Postak, J., Russell, W., Stewart, K. 1977. Software Cost Estimation Study – Study Results. Final Technical Report, RADC-TR-77-220. Doty Associates, Inc.

Jensen, R. 1983. An Improved Macrolevel Software Development Resource Estimation Model. In: Proceedings of the 5th ISPA Conference, 88–92.

Linstone, H., Turoff, M., (eds.). 2002. The Delphi Method: Techniques and Applications. Web edition, available at http://www.is.njit.edu/pubs/delphibook/.

McCall, J., et al. 1977. Factors in software quality. Vol. 1,2,3. AD/A-049-014/015/055. Nat. Tech. Inf. Service. Springfield.

Mertes, K., Ferens, D., Christensen. 1999. An empirical validation of the checkpoint software cost estimation model. Journal of Cost Analysis and Management, 35–44.

NASA, 1990. Manager's Handbook for Software Development. Revision 1. Software Engineering Laboratory Series. SEL-84-101.

NASA, 1995. Software Measurement Guide Book. Revision 1. Software Engineering Laboratory Series. NASA-GB-001–94.

NASA, 1996. Software Process Improvement Book. Revision 1. Software Engineering Laboratory Series. NASA-GB-001-95.

Nielsen, J. 1993. Usability Engineering. Morgan Kaufmann. Morgan Kaufmann Publishers, San Francisco, USA.

Parametric Estimating Initiative (PEI) 1999. Parametric Estimating Handbook. 2nd edition.

Prather, P. 1995. Design and analysis of hierarchical software metrics. ACM Computing Surveys, **27**(1): 497–518.

Putnam, L. 1978. A general empirical solution to the macro software sizing and estimation problem. IEEE Transactions on Software Engineering, **4**:345–361.

Putnam, L., Mayers, W. 1992. Measures for Excellence. Reliable Software on Time, Within Budget. Englewood Cliffs, NJ: Yourdon Press.

Rubin, H. 1983. Macroestimation of Software Development Parameters: the Estimacs System. In: Proceedings of the SOFTFAIR Conference Development Tools, Techniques and Alternatives. Arlington: IEEE Press.

Shrum, T. 1997. Calibration and Validation of the CHECKPOINT Model to the Air Force Electronic Systems Center Software Database. Unpublished masters thesis. Dayton, OH, Air Force Institute of Technology.

Sicilia, M., Cuadrado-Gallego, J.J., Crespo, J., García-Barriocanal E. 2005. Software cost estimation with fuzzy inputs: Fuzzy modelling and aggregation of cost drivers. Kybernetika, **41**(2):249–264.

Van Welie, M., van der Veer, G.C., Eliëns A. 1999. Breaking down usability: In: Proceedings of Interact 99, pp. 613–620.

Yager, R.R. 1988. Ordered weighted averaging aggregation operators in multi-criteria decision making. IEEE Trans. on Systems, Man and Cybernetics, **18**:183–190.

Juan J. Cuadrado-Gallego obtained his Ph. D. degree in Computer Science from the Carlos III University in 2001. From 2001 to 2004 he worked as a full-time lecturer at the Carlos III University, after which he joined the Universities of Valladolid and Alcalá. Also

he has taught lectures on Software Engineering at the foreign University of Rome Tre in Rome, Italy. His research interests are primarily in the areas of Software Engineering and Software Metrics. He is the Software Engineering leader of the Information Engineering Research Unit at the University of Alcalá and collaborates as editorial board member in national and international journals related to Software Engineering and Software Metrics.

Luis Fernández-Sanz obtained degree in Computer Science from the Technical University of Madrid (Spain) in 1988 and a Ph. D. degree in Computer Science from the University of the Basque Country in 1997 (with a extraordinary award mention). From 1989 to 1996 he worked as a full-time lecturer at Technical University of Madrid, after which he joined Universidad Europea de Madrid where, since 2000, is director of the department of Computer Systems. His research interests are primarily related to the areas of Software Engineering and Quality as well as educational methods and IT labour market analysis. Since 2000 is the chairman of the Software Quality Group of ATI (the main Spanish association of IT professionals) in charge of the Spanish national conference on Software Quality and Innovation (JICS). He is acted as editor, guest editor or editorial board member in different national and international journals (REICIS, NOVATICA, UPGRADE, etc.) in the area of Software Engineering.

Miguel-Ángel Sicilia obtained a Ms.C. degree in Computer Science from the Pontifical University of Salamanca, Madrid (Spain) in 1996 and a Ph.D. degree from the Carlos III University in 2003. He worked as a software architect in e-commerce consulting firms, being part of the development team of a Web personalization framework at Intelligent Software Components (iSOCO). From 2002 to 2003 he worked as a full-time lecturer at the Carlos III University, after which he joined the University of Alcalá. His research interests are primarily in the areas of adaptive hypermedia and Web personalization, learning technology, and Software Engineering. He is the leader of the AIS SIG on Reusable Learning Objects and collaborates as editorial board member in several international journals related to learning technology, metadata and the Semantic Web.