The Second Level Input Variables for Software Cost Estimation Models

Juan J. Cuadrado-Gallego[1], Javier Dolado[2], Daniel Rodríguez[3], Miguel-Angel Sicilia[4]

[1]Departamento de Informática, Universidad de Valladolid
40005, Plaza de Santa Eulalia, 9, Segovia, España

jjcg@infor.uva.es

[2]Departamento de Lenguajes y Sistemas Informáticos, Universidad del País Vasco

20009, P.M. Lardizabal, 1, San Sebastian, España

dolado@si.ehu.es

[3]Departament of Computer Science, University of Reading

Reading, RG6 6AY, UK

d.rodriguezgarcia@rdg.ac.uk

[4]Departamento de Ciencias de la Computación, Universidad de Alcalá

28871, Ctra. de Barcelona, 33.6, Alcalá de Henares, España

msicilia@uah.es

Abstract

Among the main factors to obtain successful results making software cost estimations with Parametric Mathematical Models are the definition of the equations, the revision and the correct calibration of the input parameters, and finally, but not less significant, a wise selection of the numerical values of defined parameters for each project. Ever thinking that the accuracy of the model was solved working over the first three factors, the last factor has not been researched at the same level. But, if an erroneous numerical value is chosen for the input variables, the estimations done with the model will be erroneous, no matter how fine the model have been built. This paper point out this problem and give a way to solved it by a redefinition of the parameters in two levels.

1 Introduction

Nowadays there are many methods in Software Estimation: Expertise based, Learning Oriented Based, Dynamics Based, Regression Based, Composite Based, etc. Some of them can be considered based in the use of Parametric Models or Parametric Mathematical Models, which are the most developed and widely techniques used in software estimation [1], [2], [3], [4].

These models use mathematical equations to obtain the value of a set of output dependent variables starting from the values given to another set of independent ones, and mainly four factors affect the accuracy of the estimations done with them:

1. A precise definition of the equations to be used. Thus, for example, non-lineal equations have replaced lineal ones in most of the parametric mathematical models. They are complex equations and therefore, more difficult to use. However, they offer much better results.
2. Constant refining of the parameters used. This involves not only adding or removing them to reflect changes in technology but also a thorough understanding of those selected. Thus, for example, COCOMO II [4] has eliminated some of the effort multipliers used in COCOMO 81 [5] like Execution Time Constrain (TIME) and introduced others such as Documentation Used during the Lifecycle of the Software (DOCU).
3. An accurate calibration of the numerical values for each parameter rating levels. New reviewed and enlarged data sets as well as new statistical methods like Bayesian [6] inference have been used.
4. Wise selection of the rating level for each parameter used for the selected model in order to calculate the estimations for the specific project.

2 How Parametric Mathematical Models Works

The estimations with this kind of models usually have a two step process:

1. To do a first approximation or estimation which depends on the value of a reduced set of parameters whose weight in the final result is considered greater than the rest and is not normally related to the features of the project but the product. In most of them only one variable might be used, usually the application size, which is estimated in source lines of code by means of the measurements done in any version of function points [7] [8] [9] or by any other methods like based in Graphical User Interfaces, Ballpark, Component Mapping, Sizing by module, etc [10] [3].
2. Once finished this first step, the final result is determined using another set of variables that allow the estimation to be refined by introducing the specific characteristics of the application and development environment. The input values of the rest of these parameters, which the project planning team must also calculate or select beforehand according to the specific characteristics of the projects, do not usually have a precise estimation system.

If the literature published on parametric models is revised, several studies on new mathematical equations, on cost drivers, and about how to obtain more precise numerical values for the parameters will be found. Moreover modern Software Estimation tools has resolved most of this factors.

However, for a specific software project, the choice of the most suitable rating level for each parameter is left up to the experience or aptitude of the project planning team. This team can only follow general indications or questionnaires on the generic characteristics that affect the variable in question and they must determine whether one value or another is assigned for a specific parameter.

3 The Role of the Second Level

For any parametric model, the selection of the rating level and, therefore, the adequate numerical value for the parameters is a fundamental step to make accurate estimations. This means that although the numerical value for the different rating levels of any particular parameter had been estimated with sufficient accuracy for its application in a specific project, if an inadequate rating level is selected, the estimations will be incorrect since the numerical value inputted in the model will be greater or lesser than what should correspond. Consequently, it is essential to choose the appropriate rating level for each parameter.

By other hand, the rating level of each parameter for a specific project will depend, in turn, on the rating of a set of factors or characteristics that affect the said variable. For many input variables, these characteristics are heterogeneous, that is, they reflect aspects that are essentially different. As result, for a specific project, each of these factors will have their own evaluation, independently of other factors that affect the same parameter. Consequently, each of them will have their own effect on the value chosen for the parameter and, by extension, for the final values estimated for the project.

The effect of these new parameters on the final estimation is reflected in the parameters whose value they determine and that are directly included in the equations (Figure 1). Therefore they could be considered first level parameters and termed *First Level Parameters* (FLP). This means that the new parameters can be considered second level parameters and called *Second Level Parameters* (SLP).

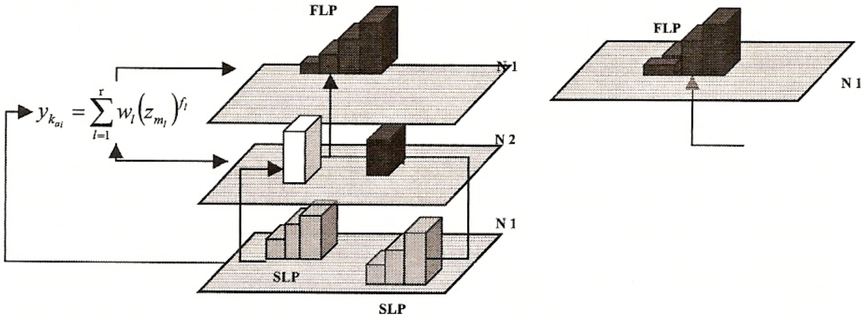

Figure 1: The two levels of input parameters

The effect of these new parameters on the final estimation is reflected in the parameters whose value they determine and that are directly included in the equations (Figure 1). Therefore they could be considered first level parameters and termed *First Level Parameters* (FLP). This means that the new parameters can be considered second level parameters and called *Second Level Parameters* (SLP).

4 An Example: The SLPs for the COCOMO II FLP DOCU

One of the FLP that is increasingly becoming important in modern software engineering practices is the documentation generated in project development. On the other hand one of the most well known and most widely used parametric mathematical models is currently the COCOMO II Post – Architecture Model, and because of its public nature and function research and development can be carried out. For these reasons the COCOMO Post – Architecture cost driver DOCU can be used as a adequate example of the meaning of the SLPs.

Very accurate estimations make by COCOMO II model are produced as a result of the new equations, the new FLP of the model, the new advanced calibration methods and the very refined and arranged data set used to calibrate them [6] [11]. These positive results mean that the different real numerical values associated with the different ranting levels proposed by the model for the cost driver documentation are accurate enough.

COCOMO II Post – Architecture Model places great importance on documentation, a range of 1.52 over the total nominal effort. However, the model presents the Table 1 for the ranting levels selection of the parameter DOCU:

Very Low	Low	Nominal	High	Very High
Many lifecicle needs uncovered	Some lifecicle needs uncovered	Correct amount for lifecicle needs	Excessive for lifecicle needs	Very excessive for lifecicle needs

Table 1: FLP DOCU ranting levels selector [4]

This table permits enough margins for the selection of one or other range and different project planners can readily choose different ones for the estimations of the same project. This may easily give rise to erroneous estimations. The use of the SLPs eliminates possible inaccuracies in the range selection of FLPs.

The following SLPs DOCU are proposed for the FLCD DOCU:

- Documentation Type (DT-DOCU)

Very Low	Low	Nominal	High	Very High
Only the basic development documentation (user requirements document, software requirements, code documents y user manual)	More refined technical development documentation which includes functional analysis and low level design.	Documentation related to software project management (project plan description, estimation document, oversight reports and final balance) in addition to the previous values	Documentation related to quality assurance: quality plans, test plans and quality registers, in addition to the previous values.	Documentation related to audits, configuration management plans and other documentation, in addition to the previous values.

Table 2: SLCD DT-DOCU ranting levels selector

- Documentation Complexity (CX-DOCU)

Very Low	Low	Nominal	High	Very High
The documentation prepared using only exits generated by CASE tools without any type of modification.	The documentation is prepared modifying (textual comments) the exits generated by CASE tools.	The content of the documentation is, in most cases, prepared by modifying the models obtained during software development.	The content of the documentation is, in most cases, a text written specifically for this purpose.	The content of the documentation is, in most cases, new models and text written specifically for this purpose.

Table 3: SLCD CX-DOCU ranting levels selector

- Use of Standards and Traceability (ST-DOCU)

Very Low	Low	Nominal	High	Very High
No standards are used to prepare the software project documentation. Documents are unrelated.	Only some technical development documents comply with a standard. There is a high-level relation among them.	All the technical development documents follow a pre-defined standard and their sections are consistent with each other.	All the technical and managerial documents follow a pre-defined standard and their sections are consistent with each other permitting the control of the status of the software project.	There are documents related to quality assurance which comply with a standard and which consistently make explicit reference to the other documents.

Table 4: SLCD ST-DOCU ranting levels selector

These levels were chosen after researching those factors that had greater direct influence on the effort used to produce the necessary documentation.

As Tables 2, 3, 4 show in detail, the documentation type refers to those aspects of development or the characteristics of the product to be documented. The complexity of the documentation is related to the technical difficulties that the development team will find when developing the documentation planned. And lastly, the use of standards and traceability refers to the use of a predefined structure for each project documentation section.

As mentioned in section 3, the basic characteristic of SLPs is that are heterogeneous; that is, the value of the FLP DOCU is obtained from the value of the three characteristics mentioned, but each of these must be studied independently.

Thus, for a specific project, the effort used to solve the complexity of the documentation may be high, while for the same project the effort used in the standards and traceability may be low. This is why the ranting levels selection for the FLP DOCU can only be accurately tackled using a method that value independently of each one of the characteristics (SLP) that will affect the variable and, subsequently, obtain a weighted result.

Thus, for example, if Documentation Type (DT-DOCU) ranting level was high (documentation related to quality assurance: quality plans, test plans and quality registers, in addition to the previous values), Documentation Complexity (CX-DOCU) ranting level would be low (the documentation is prepared modifying (textual comments), the exits generated by CASE tools). Use of Standards and Traceability (ST-DOCU) would be nominal (all the technical development documents follow a pre-defined standard and their sections are consistent with each other). (Tables 2, 3, 4) A weight result must give for the FLP a rating level nominal.

5 Conclusions

This paper presents a new method to solve the problem of wise selection of the input variables rating levels in parametric mathematical models, which is based on the simple but powerful idea of separate the input variables or parameters used by this kind of models in two levels.

The first level contains the more important parameters which accurate input is more important for a correct model function. The second level contains different input variables sets each one of them are associated to a one specific first level input variable, whose value they determine.

This method open a wide way to walk: to determine the different second variables sets for each first level one; to determine the weight of each one; to found the mathematical equations to weight each second level parameters; **etc.**

6 Acknowledgement

This research have been done under support of Spanish Ministry of Science and Technology CICYT TIC2001-1143-C03

7 References

1. Galorath Associates, SEER–SEM User's Manual (Release 4.5), Galorath Associates Technologies, El Segundo, 1996.
2. PRICE Systems, LLC, PRICE S Reference Manual (Version 3.0), Lockheed–Martin, Mt Laurel, NJ, 1997.
3. L.H. Putnam, W. Mayers, Measures for Excellence. Reliable Software on Time, Within Budget, Yourdon Press, Englewood Cliffs, NJ, 1992.
4. B.W. Boehm, C. Abts, A.W. Brown, S. Chulani, B.K. Clark, E. Horowitz, R. Madachy, D. Reifer and B. Steece, Software Cost Estimation with COCOMO II, Prentice Hall, Upper Saddle River, NJ, 2000.
5. B. W. Boehm, Software Engineering Economics, Prentice Hall, Englewood Cliffs, NJ, 1981.
6. S. Chulani, B.W. Boehm, B. Steece, From Multiple Regression to Bayesian Analysis for COCOMO II, 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), 1999.
7. IFPUG, Function Point Counting Practices Manual (Release 4.1.1), International Function Point Users Group, 2000.
8. UKSMA, Mk II Function Point Analysis Counting Practices Manual (Version 1.3.1), UK Software Metrics Association, 1998.
9. A. Abran, FFP Release 2.0: An Implementation of COSMIC Functional Size Measurement Concepts, Presentation at FESMA 99, Amsterdam, 1999.
10. J.J. Cuadrado, Software Projects Estimation Methods, Tech. Report. UC3M-TR-CS-01-1, Computer Science Department, Carlos III of Madrid University, Madrid, 2001.
11. S. Chulani, B.K. Clark, B.W. Boehm, Calibration Results of COCOMO II.1997, 22nd Software Engineering Workshop, NASA – Goddard, 1997.