Software Project Effort Estimation Non Lineal Mathematical Models

Pablo R.Soria[1], Borja Martín[1], Marian Fernández de Sevilla[1],
María J.Domínguez-Alda[1], Miguel A.Herranz[1]

[1]CuBIT, Software Measurement Laboratory; Department of Computer Science,
Unversity of Alcalá, Madrid (Spain)

pablo.rsoria@uah.es

Abstract. This paper presents a review of the software project estimation methods that have been developed across the software engineering history, mainly focused on the effort estimation non lineal mathematical models. These methods and models have been classified from new criteria and are based specifically on public dissemination models. For each model is showed its main characteristics, elements and equations that allow us to see as a whole the operation and implementation of each of these effort estimation methods.

Keywords: Software Engineering, Effort estimation, Project size, Software project planning process.

1. Introduction

Since the first computing developments until the current ones, a fundamental problem has been the fulfillment of certain deadlines for delivery within an established cost, as well as to be able to track and control the projects evolution. Therefore, the establishment of some methods that would enable us to obtain these objectives in a way as realistic and accurate as possible has become an increasingly factor for The Computing Science as a whole, and such methods have been based on knowledge acquired by different disciplines of this science, since Software Engineering to Artificial Engineering.

In addition to produce ever better results in the original objectives, the continuing evolution that the estimation methods have experienced, has also allowed to obtain other benefits such as the upgrading of the projects risk analysis or the possibility of quantitative analysis on the effectiveness of different change proposals of the software construction processes.

Since the 1960´s until now, have been published a large number of models and have also been proposed various classifications of the same based on different criteria. One of the best-known and referenced in the literature is the one proposal by Conte, Dunsmore and Shen [6] comprising four types of models:

1. *Historic / Experimental*
   It refers to those models based on a set of equations proposals by an expert.
2. *Statistics*
   It brings together the models using a regression analysis to establish the relationship between the effort and drivers of cost. It is differed two rates depending on the equations used:
   a. Lineal. Los The algorithms are lineal equations.
   b. Non Lineal. The algorithms are non lineal equations.
3. *Theoretic*
   They are based on theories on how human beings are communicated among themselves, on how the human mind works during the programming process or on mathematical laws that follows the process of developing a software product.
4. *Compounds*
   They incorporate a combination of analytical equations, statistical adjustment data and experts ideas.

The rest of this paper is structured as follows; Section 2 provides a new proposed classification for effort estimation model following the basis lines of previous models. Then Section 3 describes the non lineal and public mathematical models that we are going to present and analyze in a chronological order. Finally, conclusions are provided in Section 4.

2. New Proposed Classification for Effort Estimation Models

Here we are going to propose a new classification for effort estimation models [7] that although takes some ideas

Develop. Time:

$$t = (e)^{0.33+0.2(b-1.01)} \cdot \left(\frac{SCED\ \%}{100}\right) \qquad (14)$$

Where *t* represents the time in months, *e* represents the effort measured in MM, *b* is a constant and *SCED* is the effort multiplier Time needed for development.

3.7 COPMO

The official name for this model is *COPMO* that comes from COpeative Programming MOdel. It was first time published in 1984 by Thebaut and Shen [12].

This model uses as independent variables the size of the product to be developed and the measurement of the team size that develops the product. It is based on the following equation for Effort:

$$e = a + b.s + c(n)^d \qquad (15)$$

Where *e* represents the Effort measured in MM, *s* the size in thousands of SLOC, *a, b, c* and *d* are constants, and *n* is the average size of the development team.

4. Conclusions

In this paper, we have presented a review of the software project effort estimation methods that have been developed across the software engineering history, mainly focused on the effort estimation non lineal mathematical models. With this review, we have tried to show how these estimation models have evolved from its origins and the mathematical bases that have followed for its implementation.

Taking a clear view of how these estimation models have been based, then may understand how are developing and improving the current ones and lines we can follow to analyze the next.

This is not attempts to bring a new model or assess, it is just a simply intend to present a chronological evolution according to the new classification on software project effort estimation Non Lineal models with public dissemination that we have contributed. Therefore, as futures lines we will assist in upcoming revisions to the owner models which have been developed so far and that have led to a great progress in the improvement of the processes of effort estimation in the software project planning process.

Acknowledgement

We would like to thank the University of Alcalá for supporting this research (Ph.DC researchers support programme).

References

[1] Banker, R., Kauffman, R. and Kumar, R.: An Empirical Test of Object-Based Output Measurement Metrics in a Computer Aided Software Engineering (CASE) Environment, In *Journal of Management Information System*, 1994.

[2] Baylei, J. and Basili, V.: A Meta-model for Software Development Resource Expenditures, In *Proceedings of the Fifth International Conference on Software Engineering*, (1981), pp. 107-116.

[3] Boehm, B.: *Software Engineering Economics*, Editorial Prentice Hall, 1981.

[4] Boehm, B., Clark, B., Horowitz, E., Madachy, R., Selby, R. and Westland, C.: Cost Model for Future Software Life Cycle Processes: COCOMO 2.0, In *Annals of Software Engineering Special Volume on Software Process and Product Measurement*, Eds. J.D. Arthur, S.M. Henry and J.C. Baltzer, Edit. AG Science Publishers, Amsterdam (Holland), (1995), Vol. 1.

[5] Chulani, S., Boehm, B. and Steece, B.: Calibrating Software Cost Models Using Bayesian Analysis, En *Technical Report USC-CSE-98-508*, (1998).

[6] Conte, S. D., Dunsmore, H. E., Shen, V. Y.: *Software Engineering Metrics and Models*. Benjamin / Cumming Co., Inc. Menlo Park. 1986.

[7] Cuadrado-Gallego, J. J.: Métodos de Estimación de Proyectos Software, In *UC3M-TR-CS-2000-01* Vol. 1, Ed. Dep.Informática UC3M, (Spain), 2000.

[8] Herd, J., Postak, J., Russel, W. and Stewart, K.: Software Cost Estimation Study – Study Results, In *Final Technical Report, RADC-TR-77-220*, Ed. Doty Associates, Inc., 1977.

[9] James, T. Jr.: Software Cost Estimating Methodology. In *IEE Proceedings of National Aerospace Electronic Conference*, (1997), pp 22-28.

[10] Kauffman, R. and Kumar, R.*: Modelling Estimation Expertise in Object Based ICASE Environments*, Editorial Stern School of Business Report, New York University, January 1993.

[11] Taback, R., and Ditimore, J.: Estimation Computer Requirements and Software Development Costs, In *RM-1873*, Eds. General Research Corporation, 1974.

[12] Thebaut, S. y Shen, V.: An Analytic Resource Model for Large-Scale Software Development, In *Inf. Proc. Management*, 20 (1-2), (1984).

[13] Walston, C. and Felix, C.: (1) A Method of programming Measurement and Estimation, In *IBM System Journal*, 16, (1), (1977, pp. 54-73. (2) Author´s Response, In *IBM System Journal*, 16 (4), (1977), pp. 422-423.