Software Estimation: Universal Models or Multiple Models?

Alain Abran[1], Juan Jose Cuadrado Gallego[2]
[1]École de Technologies supérieure – Université du Québec, Montréal, Canada
alain.abran@etsmtl.ca
[2]Universidad de Alcalà de Henarès, Madrid Spain jjcg@uah.es

Abstract. In the field, there is a very large diversity of development processes in use, and various mixes of costs drivers, each with a different impact depending on the context. The classical approach to building estimation models in software engineering is to build a single estimation model and include within it as many cost factors (i.e. independent variables) as possible. In this paper, we do not postulate that there exists a single estimation model that is ideal in all circumstances, but rather we report on exploratory research conducted over the past few years looking at relevant concepts from the field of economics and from discussions with organizations attempting to understand the data that they have collected on their projects. The purpose of exploratory research is not to demonstrate a hypothesis, but to identify new potentially relevant concepts to develop hypotheses to be tested later on with empirical or experimental data.

Categories and Subject Descriptors
D2.9 **[Management]**: Productivity

General Terms
Measurement

Keywords
Process measurement, Productivity measurement, Estimation

1. INTRODUCTION

The classical approach to building estimation models in software engineering is to build a single estimation model and include within it as many cost factors (i.e. independent variables) as possible.

A- Model based on completed projects: When the builders of estimation models have access to a reasonable set of completed projects, they typically attempt to build a single model for all of these projects which takes into account the largest possible number of the variables included in their data repository. This approach is best illustrated with the design of the COCOMO models [1-3], containing a large number of cost drivers, with:

- the authors' own definition of these cost drivers,
- the authors' own measurement rules for these cost drivers and their own assignment of impact factors for each of them.

This, of course, leads to complex models with a large number of variables, but seldom with enough data points for meaningful statistical analysis or the confidence that such models can be used in environments other than the one for which they were developed initially.

B- Models based on opinions: Another approach is to build models based on the authors' opinions about the variables and their estimation of the impact on a model's behavior. With such an approach, it is very easy to come up with any number of new cost drivers: being based on opinion only, there is no cost for data collection and analysis. This can be observed in some of the 'use case points'-based models [4-7]. Furthermore, for many of such models – some available free from the web, there has not even been any attempt to demonstrate how well they perform, even within the context in which they were built.

Models built without data (or with not enough data) and those that include many opinion-based cost drivers (i.e. independent variables) lead the managers to believe that the majority of the important costs drivers have been duly taken into account by the models: the managers are then led to believe that, by using these models, they reduce the risks inherent in estimation. This makes them feel good, but falsely so, since such models are not supported by empirical evidence, and their limitations have not been documented. Moreover, lured by that 'feel good' potential, managers may find themselves dealing with even more uncertainty.

Over the past 30 years of research on software project estimation, expert practitioners and researchers have come up with many models with different mixes of cost drivers, but with little commonality, and to date most of them have not been generalized to contexts other than the one on which they were based.

In this paper, we report on exploratory research conducted over the past few years looking at relevant concepts from the field of economics and from discussions with organizations attempting to understand the data that they have collected on their projects. The purpose of exploratory research is not to demonstrate a hypothesis, but to identify new potentially relevant concepts to develop hypotheses to be tested later on with empirical or experimental data.

This paper is organized as follows. Section 2 presents a few economics concepts used to model a production process, and corresponding characteristics that may be relevant to build multiple models and interpret them. Section 3

Over the past 30 years of research on software project estimation, expert practitioners and researchers have come up with different models with different mixes of cost drivers, but with little commonality, and to date most of them have not been generalized to contexts other than the one on which they were based.

In this paper, we have reported on exploratory research looking at relevant concepts from the economics field and from discussions with organizations attempting to understand the data they have collected on their projects.

The purpose of exploratory research is not to demonstrate a hypothesis but to identify new potentially relevant concepts to develop hypotheses to be tested later on with empirical or experimental data.

In this paper, we did not postulate that there exists a single estimation model that can be considered ideal in all circumstances. Rather, we looked for concepts which could contribute to the identification of distinct models corresponding to distinct production processes.

For instance, section 2 presented a few economics concepts used to model a production process, and corresponding characteristics that may be relevant to software, such a fixed and variables costs as well as production processes with either low or high effort sensitivity to functional size. Section 3 showed another approach to the identification of distinct production models which may manifest themselves across size ranges as organizations adjust project processes as project size increases.

The authors are currently working in collaboration with industrial organizations with datasets similar to the ones discussed in this paper (wedge-shape and with different density of size ranges). Research is in progress to test the contributions of taking into account the various concepts presented in this paper for developing distinct models for the various processes identified by organizations.

Reference

[1] Boehm, B. W., *Software Engineering Economics*, Englewood Cliffs, NJ, Prentice Hall, 1981.

[2] Boehm, B. W., Abts, C., Brown, A. W.,; Chulani, S., Clark B. K., Horowitz, E., Madachy, R., Reifer, D., and Steece, B., '*Software Cost Estimation with COCOMO II*', Prentice Hall, 2000.

[3] Jørgensen, M. and M. Shepperd,. "A Systematic Review of Software Development Cost Estimation Studies", IEEE Transactions on Software Engineering 2007, Vol. 33, no. 1, pp.: 33-53.

[4] Clemmons, Roy K., "Project Estimation With Use Case Points," Crosstalk, February 2006, pp. 18-22

[5] Mohagheghi, Parastoo, "Effort Estimation of Use Cases for Incremental Large-Scale Software Development," IEEE International Conference on Software Engineering – ICSE '05, May 15-21, 2005, St.Louis, MI, USA, pp. 303-311.

[6] Nageswaran, Suresh, "Test Effort Estimation Using Use Case Points," Quality Week 2001, San Francisco, CA, USA, June 2001.

[7] Ouwerkerk, J., Abran, A., Evaluation of the Design of Use Case Points (UCP)', MENSURA2006, Conference Proceedings edited by the Publish Service of the University of Cádiz www.uca.es/publicaciones Nov. 4-5, 2006, Cadiz (Spain), pp. 83-97.

[8] Bourque, P., Oligny, S., Abran, A., Fournier, B., 'Developing Project Duration Models in Software Engineering', Journal of Computer Science and Technology, Vol. 22, no. 3, May 2007, pp. 348-357.

[9] Abran, I. Silva, L. Primera, "Field Studies Using Functional Size Measurement in Building Estimation Models for Software Maintenance," Journal of Software Maintenance and Evolution: Research and Practice, vol. 14, 2002, pp. 31-64.

[10] Kitchenham, B. A., Taylor, N. R., "Software Cost Models," ICL Technical Journal, vol. 4, no. 1, May 1984, pp. 73-102.

[11] Kitchenham, B.A., Empirical studies of assumptions that underlie software cost-estimation models. *Information and Software Technology* 34(4):211-218 1992.

[12] Lokan, C., 'What Should you Optimize when building an Estimation Model?', IEEE International Software Metrics Symposium – METRICS 2005, 2005.

[13] Abran, A., Ndiaye, I., Bourque, P., "Evaluation of a Black-Box Estimation Tool: A Case Study," In the special issue: "Advances in Measurements for Software Processes Assessment," of the journal Software Process Improvement and Practice, vol. 12, no. 2, March-April 2007, pp. 199-218.