Productivity-Based Software Estimation Model: An Economics Perspective and an Empirical Study

Alain Abran, Jean-Marc Desharnais
ETS, University of Québec
Montreal, Canada
alain.abran@etsmtl.ca
jean-marc.desharnais@etsmtl.ca

Mohammad Zarour
KAST
Saudi Arabia
mzarour@kacst.edu

Onur Demırörs
Middle East Technical University
Ankara, Turkey
demirors@metu.edu.tr

Abstract— **Management interest is not limited to accurate estimate of software projects, but also to being more productive than your peers. This paper proposes an estimation approach based on economics concepts, such as productivity models with fixed/variable costs and economies/diseconomies of scale. This paper also reports on an empirical study in a Canadian organization that illustrates this approach.**

Keywords-Software economics; productivity models; fixed and variable cost; Function Points.

I. Introduction

Over the past 40 years, researchers have tackled software effort estimation using different mixes of cost drivers as well as various techniques to combine these costs drivers using either expert opinions or mathematical models: their main goal is to come up with 'accurate estimates', either intuitively based on experts opinions, or through mathematical models, derived from distinct broad strategies for designing estimation models:

- Strategy 1: Statistical analyses taking into account only the information from completed projects. They are represented by multi-variable models with as many independent variables as there are cost drivers taken into account. Some examples are: linear and nonlinear regressions techniques, neural network models, and genetic algorithms. For an adequate statistical analysis, it is generally accepted that there should be 20 to 30 observations for each independent quantitative variable.
- Strategy 2: Statistical analyses with a unique independent variable (typically, size) combined with a single adjustment combining the impact of multiple cost drivers, which individual values come from fixed pre-determined step-functions for each cost driver. This can be observed, for instance, in the COCOMO-like models [1][2].

Multi variables models built with insufficient data points (strategy 1) or with models with an adjustment factor bundling multiple categorical variables (strategy 2) do not necessarily reduce the risks inherent in estimation: they may lead managers to believe that the majority of important cost drivers have been duly taken into account by the models but, in practice, even more uncertainty has been created.

Although accurate estimation of a single project is important, estimation is not the unique management concern, nor the most important one for a specific project or for a set of projects for an organization or a customer. For example, greater productivity, profitability, and high quality have often greater management relevance than accuracy of estimates. In contrast to the traditional approaches in software engineering focusing strictly on estimation, this paper looks at an approach common in economics which looks first at productivity, a single variable model, before moving on to multi-variable models for estimation purposes in specific contexts. Some of the concepts introduced in this paper have been explored initially in [3] to identify a new approach to software benchmarking and estimation. This paper expands on these concepts and reports on an empirical study that illustrates the contribution of these concepts from economics in developing tailor-made estimation models based on the performance of the organization studied.

The rest of this paper is organized as follows. Section II presents the productivity concept as defined in economics to represent the performance of a production process, including fixed/variable costs and economies/diseconomies of scale. Section III presents the context of an empirical study in a Canadian organization. Section IV presents the productivity analysis and the estimation models developed for this organization on the basis of economic concepts. Section V presents a summary and implication for estimation purposes.

II. Productivity Models and Economics Concepts

A. *A productivity model represents a 'production' process*

A project, on the one hand, is typically set up to plan and manage an unique event, with a start date, an end date, and a unique outcome.

On the other hand, to improve the odds of meeting the project targets, a project process is implemented to plan activities, monitor project progress, and take remedial action when something goes off track. Similarly, even though each piece of software is different, its delivery is organized in a structured manner and not left to randomness and individual moods and intuitions of the day: to deliver the right outcome on time and within the expected cost and level of quality, a 'development process' is implemented to meet the target taking into account the set of priorities and within a reasonable range of predictability.

The question is: How can the performance of a process be estimated in the future if its current and past performance

Copyright (c) IARIA, 2014.

and any variations in performance are not known? What are the economic concepts at work in software projects? And, when this is understood and quantified, how can these economics insights be used for estimation purposes?

A software development project can be modeled as a production process in its simplest form, with three main components:

1) Inputs: to calculate productivity, the people involved in the production process are considered as the inputs from an economics perspective. In a software project, the inputs are typically measured in work-hours (or person-days/-weeks/-months).
2) Activities within the process itself: for productivity calculation, all of the activities and constraints of the process are considered as a black-box and are not taken into account: they are therefore implicit variables, not explicit variables in productivity calculations.
3) Outputs: the outputs are represented by the number of functional units produced by the process. In a car manufacturing plant, the outputs of the plant are the number of cars produced (not the technical characteristics of the car, such as the weight, colors, shape, etc.). In comparison, the output of the software development process is the set of functions delivered to the users, which functions can now be quantified with international standards of measurements, such as with anyone of the relevant ISO standards on software functional size [4][5][6][7].

The productivity of a process is its ratio of outputs over the inputs used to produce such output. In software, the productivity of a software project can be represented as 10 Function Points per work-month. It is to be observed that, by convention, the productivity ratio ignores all process characteristics: it is process and technology independent and allows therefore objective comparison of the productivity of a process across technologies, organizations and time.

B. *Productivity models with fixed and variable costs*

The use of productivity models has a long history that can be traced back to a large body of knowledge developed in the domains of economics and engineering [8][9]. This section introduces some of these concepts which may also be useful in modeling, analyzing and estimating the performance of software projects.

A productivity model is typically built with data from <u>completed</u> projects, that is, it uses the information of a project for which there is no more uncertainty on:

- The outputs: i.e., all the software functions have been delivered; and,
- The hours worked on the project: i.e., they have been accurately entered into a time reporting system.

This illustrated in Figure 1 where:

- The x axis represents the functional size of the software projects completed;
- The y axis represents the effort in number of hours that it took to deliver a software project.

The straight line across Figure 1 represents a statistical model of the productivity of the software projects. More specifically, this single independent variable linear regression model represents the relationship between effort and size, and is represented by the following formula:

Y (effort in hours) = f (size)

= a x Size + b where:

- Size = number of Function Points (FP)
- a = variable cost = number of hours per function point (hours/FP)
- b = constant representing fixed cost in hours

In terms of units, this equation gives:

Y (hours) = (hours/FP) x FP + hours = hours

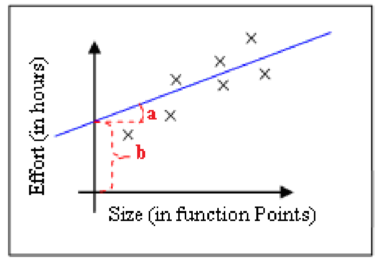

Figure 1. Fixed and variable cost in a productivity model

Insights from economics have identified two distinct types of costs incurred to produce different quantities of the same types of outputs:

<u>Fixed costs</u>: the portion of the resources expended (i.e., inputs) that does not vary with an increase in the number of outputs. In Figure1, this corresponds to b, the constant in hours at the origin when size = 0.

<u>Example of a *fixed* cost</u>: a cost of b hours of project effort is required for mandatory project management activities, whatever the size of the software to be developed.

<u>Variable costs</u>: the portion of the resources expended (i.e., inputs) that depends directly on the number of outputs produced. In Figure 1, this corresponds to the slope of the model, that is: slope = a in terms of hours/FP (i.e., the number of work hours required to produce an additional unit of output).

It is to be observed that in productivity models, the constant b does not represent the errors in the estimates as in multi-variable estimation models: in productivity models, it has a practical interpretation corresponding to the economics concepts explained above, that is: the portion of the cost that do not vary with increases in the production outputs.

C. *Economies and diseconomies of scale in productivity*

In economics, various behaviors in productivity have been observed as the number of outputs increases. For instance, that are some processes where:

- As output increases, 1 additional unit of output requires a <u>smaller</u> increase of inputs, and

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-367-4

- As output increases, 1 additional unit of output requires a greater increase in input.

When the increase in output units requires a correspondingly smaller increase in the number of input units, the production process is said to have lower sensitivity to size increases: this is referred to as 'economies of scale' (i.e., the larger the number of units produced, the more productive the production process).

By contrast, when an increase in output units requires a larger increase in the number of units for each additional output, then the production process is said to have diseconomies of scale (i.e., it is highly sensitivity to increases in size: for each additional unit produced, the less productive the production process = diseconomies of scale).

The next question is, of course, what cause these different behaviors? Of course, the answers cannot be found by graphical analysis alone, since in productivity there is only a single independent quantitative variable in a two-dimensional graph. This single independent variable does not provide, by itself, any information about the other variables, or about similar or distinct characteristics of the completed projects for which data are available. Efficiency investigation with additional independent variable can help identify which other variables cause variations in productivity and to which extent for each.

When a data set is large enough (that is, 20 to 30 data points for each independent variable), the impact of the other variables can be analyzed by statistical analysis. In practice, most software organizations do not have data set large enough for valid multi-variable statistical analysis. However, within a single organization the projects included within a data set can be identified nominally by the organizations that collected the data [3][10]. Each project in each subset should be analyzed next to determine:

- Which of their characteristics (or cost drivers) have similar values within the same subset; and
- Which characteristics have very dissimilar values across the subsets.

Of course, some of these values can be descriptive variables with categories (i.e., on a 'nominal' scale type: for example, a specific Data Base Management System (DBMS) has been used for a subset of projects, etc.). It then becomes necessary to discover which additional independent variables have the most impact on the relationship with project effort. The different values of such characteristics can then be used to characterize such datasets, and for selecting which of these productivity models to use later on for estimation purposes.

III. A PRACTICAL USE OF THESE ECONOMIC CONCEPTS: AN EMPIRICAL STUDY

A. *Context*

A Canadian organization was interested in determining its own productivity, in understanding some of the key drivers behind its major productivity variations, and in using the findings to improve its estimation process.

This organization, a government agency, provides specialized financial services to the public, and its software applications are similar to those of banking and insurance providers. It has a software development methodology fully implemented across all of its projects. The main objectives of this empirical study were to:

1. Internal benchmarking, i.e., compare the productivity of individual projects.
2. Develop estimation model(s) based on the data collected.
3. Identify and explain significant productivity variations across their projects.

B. *Data collection procedures*

The initial step was to identify the projects that could be measured for the productivity and benchmarking analyses. The selection criteria were:

- Projects completed within the previous two years, and
- Project documentation available for functional size measurement.

For this study, all data were recorded using the data field definitions of data collection questionnaire of the International Software Benchmarking Standards Group [11] [12].

C. *Data Quality Controls*

Quality control of the data collection process is important for any productivity study. Here, two quantitative variables are critical: the effort reported for each project, and the project functional size:

A- Effort data: in this organization, the time reporting system is considered highly reliable and is used for decision making, including payment of invoices when external resources are hired to complement project staffing.

B- Measurement of functional size: the quality of the measurement results depends on the expertise of the measurers and on the quality of the documentation available for the measurement process. For this productivity study, all functional size measurements were carried out by the same measurer with years of 20 years expertise in both functional size measurement methods used.

D. *Descriptive Analysis*

For this study, the 16 software development and improvement projects completed between 2004 and 2006 were measured in terms of functional size, effort, and various environment qualifiers. The staff who developed

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-367-4

these projects included both internal and external developers, distributed equally overall. In summary:

- Project sizes vary from a minimum of 111 FP (project 7) to a maximum of 646 FP (project 2).
- Effort varies from 4,879 hours to 29,246 hours.
- Unit effort varies from 14 hours/FP for project 12 to up to 98 hours/FP for project 6, a factor of approximately 8 between the least productive and the most productive within the same organization.
- Duration varies from 10 to 35 months.
- Maximum development team sizes for 12 of the 16 projects were available, and ranged from 6 to 35 employees.

The descriptive statistics of this dataset are as follows:

- Average effort = 12,033 hours (or, 1,718 person-days at 7 hrs per day, or 82 person-months at 21 days per month).
- Average unit effort is 41.5 Hrs/FP
- Average duration = 18 calendar months.

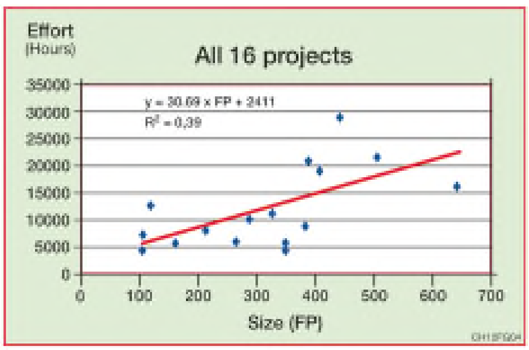

Figure 2. The organization's overall productivity model – N = 16 projects

IV. PRODUCTIVITY ANALYSIS AND ESTIMATION MODELS

A. *The overall productivity model for this organization*

The dispersion of points for this organization is illustrated in Figure 2 for all 16 projects, with functional size on the x axis, and effort on the y axis: it shows also the overall single-variable productivity model for this organization, using a single regression model:

Effort = 30.7 hours/FP x project size + 2,411 hours

The coefficient of determination (R^2) of this model is relatively low, at 0.39.

The practical interpretation of the above equation is as follows for this organization:

- Fixed effort = 2,411 hours
- Variable effort = 30.7 hours/FP

The possible reasons for the rather high fixed and high variable unit effort figures have been discussed with the managers, and the following observations provided in terms of the development methodology deployed in this organization:

A. It is highly procedural and time-consuming;
B. It included heavy documentation requirements;
C. It requires lengthy consensus building procedures across stakeholders and development staff;
D. It requires a relatively high number of inspections.

From Figure 2, it can be observed that, for this organization, 5 projects have costs 100% higher than projects of comparable functional size:

- Project with 126 FP required twice as much effort as 2 other projects of similar size.
- Four large projects (between 400 and 500 FP) required two or three times as much effort as similarly sized projects: the effect of these projects is to pull up the linear model (and corresponding slope) and to influence both the fixed and variable costs considerably.

This data sample was therefore split into two groups for further analysis.

A. A group of 11 projects which have the best productivity (i.e., lower unit effort, and which are below the regression line in Figure 2).
B. A group of 5 projects which have a much worst productivity (i.e., a unit effort twice the unit effort of the 11 other projects, and which are above the regression line in Figure 2).

B. *Organizational process capability: the most productive projects*

A productivity sub-model is built next with the 11 projects with a much lower unit effort per project that is, the most productive ones. For these projects, the linear regression model is:

Effort = 17.1 hours/FP x size of the project + 3,208 hours

The coefficient of determination (R^2) of this model is 0.589, higher, relatively, than that for the overall model.

The practical interpretation of this equation is:

- Fixed costs = 3,208 hours
- Variable Costs = 17.1 hours/FP

C. *Productivity model of the least productive projects*

Another productivity sub-model is built with the 5 least productive projects in group B. For these projects, the linear regression model is:

Effort = 33.4 hours/FP x project size + 8,257 hours

The coefficient of determination (R^2) of this model is better, at 0.637. Of course, with a sample of only five projects, this number is not statistically significant, but is still interesting for this organization.

The practical interpretation of the above equation is as follows:

- Fixed effort = 8.257 hours
- Variable effort = 33.4 hours/FP

This group of the five least productive projects is characterized by a fixed cost which is almost 4 times higher than that of the full set of projects (8,257 hours vs. 2,411 hours), and a relatively similar variable effort unit (33.4 hours/FP vs. 30.7 hours/PF).

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-367-4

The group of 11 most productive projects is characterized by a fixed cost which is approximately 40% lower than that of the least productive projects (3208 hours vs. 8257 hours), and a variable unit effort which is almost 50% lower (17.1 hours/FP vs. 32.4 hours/FP); that is, with interesting economies of scale and an R^2 of 0.55.

A summary of each group is presented in Table I, where these 11 projects represent the organization's 'capability' to deliver in normal conditions and the other five projects illustrate how projects are significantly impacted in the presence of factors which have not yet been identified through this single independent variable (i.e., functional size) analysis. Exploration of these additional impact factors is discussed in Section V.

D. *Qualitative causal analysis*

Of course, a single independent variable model cannot explain the causes of such variations. Furthermore, there are often not enough data points within a single organization (unless they have been collecting data for many years) to rely on quantitative analysis with a dataset of only sixteen projects: each additional independent typically requires 20 to 30 additional data points. In the absence of sample sizes large enough for quantitative analysis, qualitative analysis can help identify probable causes of increases. In the context here, qualitative analysis will not attempt to quantify precisely the impact of a cause (or cost drivers), but will attempt to identify qualitatively which factor could have had the greatest negative impact on productivity.

TABLE I. FIXED AND VARIABLE EFFORTS: CAPABILITY VERSUS LEAST PRODUCTIVE PROJECTS

Samples/ Regression coefficients	**All 16 projects**	**Most productive: 11 projects**	**Least productive: 5 projects**
Fixed effort (hours)	2,411	3,208	8,257
Variable effort (hours/FP)	30.7	17.1	34.4

Off hand in the causal analysis of the productivity variations in this organization, two candidate cost drivers were eliminated since they were considered as constant in both groups of productivity performance:

- Development methodology: in this organization the use of the industry-tailored development methodology is fully deployed across all software development projects: none of the activities and controls can be bypassed. Therefore, there was no development methodology difference across all projects.
- Project managers' expertise: some of the projects managers had, within this same 2-year period, supervised projects which were both among the most productive and the least productive. Therefore the project management expertise of specific project managers could not explain large project productivity differences.

The question is, what are the factors that led to such large (i.e., +100%) increases in unit effort? What could have been the major cause-effect relationships? To identify and investigate these relationships, available project managers were interviewed to obtain their feedback on what they believed had contributed to either an increase or a decrease in the productivity of their respective projects. The project managers interviewed had managed 7 of the 16 projects:

A. 3 projects with the lowest productivity;
B. 2 projects with average productivity;
C. 2 projects with the highest productivity.

The aim of the interviews was to obtain qualitative information from the project managers on the factors they believed had contributed, or not, to the increase in project effort compared to that of other projects of similar size developed in the organization's environment or elsewhere during their project management practice. Their feedback is summarized in the following factors:

A- The most productive projects had the following characteristics:

1. Users familiar with both the business and software development processes;
2. Users involved throughout the project;
3. Software developers working on the projects who were experienced in the use of the development environment.

B. The least productive projects had the following characteristics:

B1. Customer related issues:

1. Customer requirements that were poorly expressed, or a customer representative who did not know his environment (business area), leading to frequent change requests during a project life cycle.
2. High turnover of users involved in the projects, leading to instability in the requirements and delays in decision making.
3. Customers not familiar with the software development process in the organization, including their required involvement in project activities, including activity reviews.

B2. Project constraints:

1. Tight project deadlines for legal constraints or public face-saving that led to compressed schedule and resources being piled up to make the problem disappear.
2. New technologies unknown to the developers.

B3: Product constraints:

1. Multiple links with other software applications of the organization.

An example of negative product constraint was reported for the project with the highest unit effort (98 hours/FP): the software delivered by this project was of a small functional size, but required twice as much effort to develop as another

Copyright (c) IARIA, 2014.
ISBN: 978-1-61208-367-4

of software of similar size because it interacted with almost all the other software applications of the organization and was dependent on other organizational units. Another project had a very tight deadline, which led management to 'throw' resources at the problem to meet the deadline irrespective of the total effort required.

It can be observed that, although it was possible to identify 'qualitatively' some factors with major negative impact, the sample size was much too small for statistical tests to quantify such an impact.

V. Summary and Implications for Management and Estimation Purposes

Taking into account the related performance concepts from the field of economics, including fixed/variable costs and economies/diseconomies of scale, this paper has reported on the productivity analysis of software projects developed by a governmental organization. For this organization, three productivity models were identified which represented respectively:

- An overall productivity of this organization. This overall productivity model will be used later across times periods to verify whether or not the productivity of this organization is improving over time, and with respect to external similar organizations.
- A productivity model built from the best productive projects: it exhibit economies of scale in the development process of this organization and represents its capability to deliver a software project with a lower fixed/variable effort structure, in the absence of major disruptive factors.
- A productivity model based on the five projects with the highest unit effort: in this organization, the presence of disruptive factors has led to greater than 100% increase in project effort in comparison to their organizational process productivity capability.

Of course, the limited number of projects available in these mathematical models does not permit generalization to other contexts, but it is describing quantitatively and objectively productivity facts: these models are representative of the organization studied in which a unique software development methodology is widely implemented and represents well deployed corporate software practices, not varying individual practices (i.e., a repeatable process rather than unpredictable individual and ad-hoc practices).

For estimation purposes, the organization's process capability model represented by the best performing projects should be used, provided that a risk analysis has not detected the presence of any of the disruptive factors that have in the past increased effort twofold in this organization. Whenever such disruptive factors are identified with a high probability of occurrence within an estimation context, it justifies this organization to estimate such projects using the productivity model derived from the least productive projects. The use of these two single-variable productivity models would be expected to provide more accurate estimates that the overall productivity model combining all previous projects.

In addition, interviews with project managers allowed to identified, qualitatively for this specific organization, factors having impacted, positively or negatively, productivity, (such as: customer related issues, project constraints and product constraints): these factors were integrated next as risk factors into their revised estimation process.

This context of an organization having measured only a small set of projects is representative of many organizations without much historical data: this is a context where there are not enough data points to build with high confidence multi-variable estimation models representing local conditions and related organizational performance.

The insights from productivity models developed from an economic perspective are important since relevant improvement actions may directly impact the productivity of the organization, by lowering either of the fixed or variable project costs.

References

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

[2] B. W. Boehm *et al.* Software Cost Estimation with COCOMO II, Prentice Hall, 2000.

[3] A. Abran and J. J. Cuadrado, "Software Estimation Models & Economies of Scale," 21st International Conference on Software Engineering and Knowledge Engineering - SEKE'2009. Boston (USA), July 1-3, 2009, pp. 625-630.

[4] ISO/IEC 19761: Software Engineering – COSMIC - A Functional Size Measurement Method. International Organization for Standardization, Geneva, 2011.

[5] ISO/IEC 20926: Software Engineering - IFPUG 4.1 Unadjusted functional size measurement method - Counting Practices Manual. International Organization for Standardization, Geneva, 2009.

[6] ISO/IEC 24750: Software Engineering - NESMA functional size measurement method version 2.1 - Definitions and counting guidelines for the application of Function Point Analysis, International Organization for Standardization, Geneva, 2005.

[7] ISO/IEC 20968: Software Engineering - Mk II Function Point Analysis - Counting Practices Manual. International Organization for Standardization, Geneva, 2002.

[8] N.G. Mankiw, Principles of Microeconomics, Ed. South-Western Cengage Learning, Mason (OH) 2014.

[9] S.T. Hackman, Production Economics - Integrating the Microeconomic and Engineering Perspectives, Springer-Verlag Berlin Heidelberg, 2008.

[10] A. Abran, I. Silva and 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.

[11] ISBSG R11, International Software Benchmarking Standard Group, 2009 URL: http://www.isbsg.org/ last accessed date: July 26, 2014.

[12] L. Cheikhi, A. Abran, and L. Buglione "ISBSG Software Project Repository & ISO 9126: An Opportunity for Quality Benchmarking," UPGRADE vol. VII, no. 1, 2006, pp. 46-52.