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
presents next an approach to build distinct models by size ranges.

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

2. PRODUCTION MODELS: SOME CHARACTERISTICS

2.1 A production process

How can the performance of a development process be estimated in the future if its current and past performance, and the variations in that performance, are not known?

- What is a development process?
- How is its performance determined?
- How can a development process be modeled to build estimation models?

A development process can be modeled as a production process: this can be illustrated in its most simplified form with three main components – see Figure 1: Inputs, Activities within the process itself (in software, this corresponds to the development life cycle selected and implemented), and Outputs (the software itself and the environment in which it will be executed).

![Figure 1: A production process](image)

The inputs to a production can typically be classified into three groups:

1- The objectives for a specific production run: In software engineering, this corresponds to the objectives of the software development project. These objectives are often stated in terms of functional requirements and non-functional requirements for the software to be delivered, as well as in terms of the project priorities (costs, quality, duration).

2- The human resources to be made available for the development process, that is, the staff who will be available to work on the development project and carry out all the tasks of the sub processes within the process. These inputs are typically measured in work-hours (or person-days/weeks/months).

3- Any other non-human resources available for carrying out the tasks, such as the technical environment, including the hardware-software platform, the tools, the methodologies, etc.

2.2 Productions models with fixed and variable costs

A production model is typically built with data from projects completed, that is, when:

- all the information on a project is available;
- there is no more uncertainty on either the inputs or the outputs: all the software functions have been delivered; and
- all the hours worked on the project have been accurately entered into a time reporting system.

The points on the graph in Figure 2 represent, then, the number of hours it took to deliver the corresponding functional size of the projects completed.

- 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 2 represents a statistical model of the production process and is based on the performance of past projects.

This linear model models the relationship between effort and size, and is represented by the following formula:

\[ Y \text{ (effort in hours)} = f(x) = a \times \text{Size} + b \]

where:

- \(a\) = variable cost = number of hours per Function Point (hours/FP)
- \(b\) = fixed cost in hours

In terms of units, this equation then gives:

\[ Y \text{ (hours)} = (\text{hours/FP}) \times \text{FP} + \text{hours} = \text{hours} \]

In a production process, there are typically two major types of costs incurred to produce different sets of the same types of outputs:

**Fixed costs**: the portion of the resources expended (i.e. inputs) that does not depend on the number of outputs. In Figure 2, this corresponds to \(b\), the constant in hours at the origin when \(x = 0\).

- **Example of a fixed cost**: at \(x = 0\), \(b\) represents the fixed cost of this production process (i.e. in this organization, a cost of \(b\) hours of project effort is required to set up and manage a project independently of its size).

**Variable costs**: the portion of the resources expended (i.e. inputs) that depends directly on the number of outputs produced. In Figure 2, this corresponds to the slope of the model, that is: slope = \(a\) in terms of hours per function point.
(that is, the number of work hours required to produce an additional unit of output, that is, the independent variable x).

Effort (in hours)  
Size (in Function Points)

Figure 2: Production model: fixed & variable costs

2.3 Wedge-shaped datasets

A graphical representation of project datasets in software engineering typically has the wedge-shape distribution illustrated in Figure 3 [8-9]. It can be observed in this figure that, as the project effort increases on the x axis, there is a corresponding larger dispersion of the data points across the vertical axis: for projects of similar effort, there are increasingly wide variations of project duration on the y axis as the project effort increases. This is often referred to as a wedge-shaped dataset. This had initially been observed by [10-11], and it is typical of most data subsets built with data from large repositories (such as illustrated in [8]).

Effort (in hours)  
Size (in Function Points)

Figure 3: Example of a wedge-shaped dataset [8].

2.4 Low & high sensitivity to functional size: multiple models?

In production processes, there might be ones where:
- 1 additional unit of output requires exactly 1 additional unit of input,
- 1 additional unit of output requires less than one additional unit of input, and
- 1 additional unit of output requires more than one additional unit of 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: 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 high sensitivity to size: for each additional unit produced, the less productive the production.

Let us revisit the typical pattern of wedge-shaped datasets of software projects – see Figures 3 and 4. When looked at with the analytical grid of the concepts of low and high sensitivity to size, this single wedge-shaped dataset can be decomposed into three subsets, as follows – see Figure 4:

- **Zone 1**: The lower part of the wedge-shaped dataset. This lower part represents the set of projects demonstrating little sensitivity to increases in size: indeed, for this subset, even large increases in size do not lead to noticeably correspondingly large increases in effort. In practice, it is as if, in this subset, the effort required is almost insensitive to an increase in the number of functions in the software being developed.
- **Zone 3**: The upper part of the wedge-shaped dataset. This upper part represents the set of projects demonstrating high sensitivity with respect to functional size as the independent variable (that is, a small increase in size requires a much larger increase in effort – in either fixed or variable costs, or both).
- **Zone 2**: Finally, there is sometimes a third dataset that is somewhere in the middle range of the wedge-shaped dataset.

This leads, then, to three distinct production models (often referred to as ‘estimation models’ in the software engineering literature):

\[ f_1(x) = a_1x + b_1, \]

which corresponds to a data sample in zone 1.
\[ f_2(x) = a_2 x + b_2, \] which corresponds to a data sample in zone 2.

\[ f_3(x) = a_3 x + b_3, \] which corresponds to a data sample in zone 3.

Each of these 3 models has its own slope \((a_i)\), as well as its own fixed costs \((b_i)\). The next question is, of course, what causes these different behaviors?

Of course, the answer cannot be found by graphical analysis alone, as there is only a single independent variable in a two-dimensional graph. This single 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.

However, the projects included within each subset can be identified nominally by the organizations having collected such data [9]. Each project within each subset should next be analyzed to figure out:

- which of their characteristics (or cost drivers) have similar values within the same subset; and
- which characteristics have very dissimilar values across the 2 (or 3) subsets.

Of course, some of these values can be categories (on a 'nominal' scale type: for example, a specific Data Base Management System (DBMS) has been used for a subset of projects, etc.).

The ability to discover the different values of such characteristics can then be used to characterize such datasets, and to set the parameters for selecting which of these three production models to use later on for estimation purposes.

**3. DISTINCT MODELS BY SIZE RANGE**

We have often observed in organizations measuring the size of their projects the following:

- a large number of small projects, a smaller number of medium size projects and, in comparison, few much larger projects [12].
- a lightweight development process for small projects, a very heavyweight development process for very large projects and a tailored process for the medium-size projects.

Notwithstanding this, the usual approach in software engineering is to look for a single model across all the size ranges. An alternative approach is to build estimation models by segregating the available dataset into ranges of project sizes when an organization has different processes by size ranges.

When such information about the organizational processes is not known, a similar analysis can be done by segregating by the data set by density distributions in contrast to the general practice of building a single model for the whole range of data points [13]. Indeed, a single model across the full range is not a strict requirement of the statistical techniques used to build such models; if this is the case, users should beware, since the levels of confidence may vary across the ranges when the data is not normally distributed).

These concepts are illustrated in Figure 5 with an exemplary data set with a large number of small projects (e.g. fewer than 100 CPF), while there are only a few projects in the much wider interval for large projects. Of course, a single estimation model can be built with this full dataset. However, another approach would be to build multiple regression models representative of the density of the data points that can be identified graphically [13].

![Figure 5: Dispersion of data points](image)

For example, in Figure 5, there are:

- about 20 small projects within the 15 to 150 FP range
- about 10 projects within the 200+ to 600 range
- only 3 projects within the 1,000 to 1,300 FP range

For this specific dataset, it would be much better to build two estimation models, one for the very small projects (Figure 6) and one for the mid-range projects (Figure 7), and to consider the 3 largest projects as an analogy base without statistical strength. This would be preferable to building a single estimation model over such a large range of projects with distinct project densities, and more representative of the projects themselves.

For the small projects of this illustrative set of Figure 5, the regression equation is (Figure 6):

\[ \text{Effort} = 1.01 \times \text{FP} + 3 \text{ with an } R^2 \text{ of 0.87}. \]
For these small projects, the fixed cost is low, that is = 3, and the variable cost is close to 1.

For the mid-sized projects (within this illustrative dataset, of course), the equation is (Figure 7):
Effort = 0.32*FP + 192 with an R^2 of 0.59, but with a much higher fixed cost of 192 and a lower slope of 0.32, instead of the steeper variable cost of 1.01 for the small projects. Also, with a slope of 0.32, it exhibits much lower sensitivity to an increase in size within that size range than much smaller projects.

\[ y = 0.326x + 192 \]
\[ R^2 = 0.60 \]

Figure 7: Regression model for the 200 to 600 FP interval

For example, for the estimation of a small project of 50 FP, equation A is highly preferable to the general equation:
- Equation A is built with small projects only, and is therefore much more representative of the one being estimated. This equation is not influenced either by the very small projects or by the much larger projects. Caution must still be exercised, however, because of the limited number of projects within that range to build the estimation model for this range.

For the estimation of a mid-range project of 500 FP, equation B is preferable to the general equation:
- Equation B is built with mid-range projects only, and is therefore much more representative of the one being estimated. This equation is not influenced either by the very small projects or by the much larger projects. Caution must still be exercised, however, because of the limited number of projects within that range to build the estimation model for this range.

For the estimation of a very large project of 1000 FP, there is no generalization significance to Equation B, although these 3 data points can still be used for analogy purposes.

For purposes of comparison, the single model with the single equation for the full dataset is presented in Figure 8. For the full set of projects, the equation is Effort = 0.748*FP + 22 with an R^2 of 0.967.

\[ y = 0.748x + 22 \]
\[ R^2 = 0.967 \]

Figure 8: Model with the full dataset

While this model appears to have a better R^2 of 0.967, it is influenced too much by the three largest projects and is not representative of the majority of much smaller projects; therefore, calculation of the magnitude of the relative error (MRE) would lead to a larger MRE for the model of the full dataset, and smaller ones for the models per subset of projects within the size intervals identified.

Also, for the full dataset, the normality distribution of the data is not met, and its regression is correspondingly less statistically meaningful. By contrast, within the two size intervals identified, they would be closer to a normal distribution – within their ranges, of course.

4. SUMMARY
In the field, there is a very large diversity of development processes, and different mixes of costs drivers, each with a different impact depending on context.
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 as 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


