The Journal of Systems and Software 85 (2012) 2494–2503

Contents lists available at SciVerse ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Interpretation problems related to the use of regression models to decide on economy of scale in software development

Magne Jørgensen[a,b,*], Barbara Kitchenham[c]

[a] Simula Research Laboratory, P.O. Box 134, 1325 Lysaker, Norway
[b] University of Oslo, P.O. Box 1072, 0316 Oslo, Norway
[c] Keele University, Keele, Staffordshire ST5 5BG, UK

ARTICLE INFO

Article history:
Received 6 September 2011
Received in revised form 22 May 2012
Accepted 22 May 2012
Available online 23 June 2012

Keywords:
Software economics
Economy of scale
Regression analysis

ABSTRACT

Many research studies report an economy of scale in software development, i.e., an increase in productivity with increasing project size. Several software practitioners seem, on the other hand, to believe in a diseconomy of scale, i.e., a decrease in productivity with increasing project size. In this paper we argue that violations of essential regression model assumptions in the research studies to a large extent may explain this disagreement. Particularly illustrating is the finding that the use of the production function ($Size = a \cdot Effort^b$), instead of the factor input model ($Effort = a \cdot Size^b$), would most likely have led to the opposite result, i.e., a tendency towards reporting diseconomy of scale in the research studies. We conclude that there are good reasons to warn against the use of regression analysis parameters to investigate economies of scale and to look for other analysis methods when studying economy of scale in software development contexts.

1. Introduction

Economy of scale is a term commonly used in production industries to denote a reduction in cost per unit produced as the quantities of production inputs increases. One frequently used production function for the analysis of economy of scale is the Cobb–Douglas production function (Cobb and Douglas, 1928). The original Cobb–Douglas production function is in the form $Y = a \cdot X_1^b \cdot X_2^c$, where X_1 is the labor input, X_2 the capital input and Y the quantity of produced output. Using the Cobb–Douglas production function, we have an economy of scale if $b + c > 1$, diseconomy of scale if $b + c < 1$, and constant return on scale if $b + c = 1$. Interestingly, in software engineering we use the reverse relation to decide on economy of scale, i.e., we study a function on the Cobb–Douglas format of the type $X = a \cdot Y^b$. This model is typically formulated as $Effort = a \cdot Size^b$ and log-transformed to the linear version $\ln(Effort) = \ln(a) + b \ln(Size)$ to ease the calculation of the b-value using least square linear regression analysis. The use of a factor input (effort) function means that we have an economy of scale if $b < 1$, a diseconomy of scale if $b > 1$ and a constant return to scale if $b = 1$.

The interest in scale economies in software development seems to be strong. A search (May 3, 2012) in Google Scholar with the terms ("economy of scale" OR "diseconomy of scale") AND "software development" gives, for example, more than 1000 hits. The topic is not only interesting to better understand the nature of software development, but is also of potential relevance to software practitioners. If there is an economy of scale, this is an argument for a manager to try to reduce the software development effort by joining smaller projects into larger ones. If there is a diseconomy of scale, the manager may on the other hand try to reduce the effort by splitting larger projects into smaller projects or deliveries, e.g., through incremental development models. It may also be relevant as input to judgments and decisions related to planning and to optimal usage of resources.

It has been much debated among researchers whether and when there is a tendency towards economy of scale, diseconomy of scale or constant return of scale in software development, see (Kitchenham, 2002) for some elements of this discussion. The basis of this debate is frequently the variation of the reported b-values calculated using regression analysis of log transformed effort and size data. In the survey of twelve software development studies reported in Dolado (2001) there are b-values varying from 0.66 to 1.49. Eight of the twelve studies found $b < 1$, which may reflect a tendency towards reporting economy of scale in software development contexts. A strong economy of scale is, as far as we have experienced, in conflict with what many software professionals would consider likely. Consider, for example, a b-value of 0.8, which is not unusual to report for software development data sets. This b-value implies that as the software size gets ten times larger, e.g., from 1000 to 10,000 lines of code, we will need only 6.3 times more effort ($10^{0.8} = 6.3$). When the software size

* Corresponding author.
E-mail address: magnej@simula.no (M. Jørgensen).

0164-1212/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2012.05.068

gets 100 times larger, e.g., from 1000 to 100,000 lines of code, we will need only 40 times more effort ($100^{0.8} = 40$). With the possible exception of some types of maintenance environments (Banker and Slaughter, 1997), most software professionals seem to believe in a diseconomy rather than an economy of scale. We conducted an informal review of software project planning advice published on the internet and found several indications of a belief in diseconomy of scale and none in economy of scale. McConnel (2004), for example, claims that one should plan to spend an increasing proportion of effort on non-programming activities with increasing project size. He also argues that the programming (coding) productivity is close to constant with increasing software size in lines of code. If the proportion of non-programming activities increases and programming productivity is constant, we will observe a diseconomy of scale. Similarly, Jones (1991) claims that one should expect that management and support effort increases as the size of the project increases. A diseconomy of scale is also in accordance with results reporting increasing administrative overhead with increasing organization size, e.g., Jamtveit et al. (2009).

The dominance of researchers reporting economy of scale in software development based on analysis of software data sets, in spite of the belief in diseconomy of scale among several software professionals, is a motivation for the analyses presented in this article. Software practitioners and researchers probably use different strategies to reach their conclusions about scale economies. Software practitioners may for example rely on their experience with increase in percentage effort used on administration and testing as the project increase in size. Software researchers, on the other hand, seem to rely much more on regression analyses of the relation between development effort and software size. As we will try to show in this article, regression analysis may not be suited for deciding on scale economies in software development and the researchers' use of regression analysis-based parameters to assess economy of scale may have led them to report economy of scale in situations where the underlying relationship is linear or even diseconomy of scale.

It is important to note that this paper is *not* about building estimation models, nor is it about identifying the "true" scale economy in software development. The main aims of the papers are to:

1. Explain why software engineering data sets on effort and size, analyzed using linear regression, seem to be dominated by economies of scale.
2. Explain why there are problems interpreting the estimate of the exponential term in an effort-size regression model, such as those used for effort estimation, as an indicator of scale economy or diseconomy.

The remaining part of the paper is organized as follows: Section 2 examines ten software development data sets and finds several instances of interpretation problems, e.g., data sets where we simultaneously find economy and diseconomy of scale dependent on whether we use the production function (regression of size on effort) or the factor input (regression of effort on size) model. We use the interpretation problems as a first step to suggest that there are severe problems with the use of the *b*-value of an effort-size regression model as indicator of economy or diseconomy of scale. Section 3 discusses three reasons for the interpretation problems: the role of random error in the independent variable, incompletely specified models and non-random sampling. Section 4 briefly discusses alternative strategies to assess scale economies in software development. Section 5 concludes.

2. Simultaneously economy and diseconomy of scale

The common model of the relation between software development effort (*Effort*) and size (*Size*) is based on the following factor input function format:

$$Effort = a_1 Size^{b_1} \quad (1)$$

The production function of the same relationship is the reverse model:

$$Size = a_2 Effort^{b_2} \quad (2)$$

In the deterministic case, we may reformulate (1) to:

$$Size = \left(\frac{Effort}{a_1}\right)^{1/b_1},$$

$$\text{which shows that } b_2 = 1/b_1 \text{ and } a_2 = (1/a_1)^{1/b_1} \quad (1a)$$

If $b_1 > 1$, we have a diseconomy of scale and we must also have $b_2 = 1/b_1 < 1$. If $b_1 < 1$ we have an economy of scale and we must also have $b_2 = 1/b_1 > 1$. However, if either *Size* or *Effort* or both variables are subject to random error, the parameters of these models can be estimated using linear regression after we log-transform them:

$$\ln(Effort) = \ln(a_1) + b_1 \ln(Size) \quad (3)$$

$$\ln(Size) = \ln(a_2) + b_2 \ln(Effort) \quad (4)$$

The logarithmic transformation is not simply there to allow the value of the *b*-parameters to be estimated using linear regression, although it is important to note that there is no simple closed form algorithm to directly calculate the parameters in (1) or (2). The logarithmic transformation is a normalizing transformation that also reduces the impact of atypical data points. This is needed because effort and size data are not usually normally distributed. A further advantage of the transformation of a normal distribution is that standard linear regression can be used to test whether or not the estimated *b*-parameter is significantly different from 1 and to calculate the confidence interval of the estimate.

Notice also that the analysis of increasing (economy of scale) or decreasing (diseconomy of scale) productivity with increasing size applying the model $Productivity = Size/Effort = a \cdot Size^b$ is the same as the analysis based on the above models (3) and (4). This is the case since $\ln(Size/Effort) = \ln(Size) - \ln(Effort) = \ln(a) + b \cdot \ln(Size)$, which can be transformed to the format of for example model (3), i.e., to the model $\ln(Effort) = -\ln(a) + (1 - b)\ln(Size)$.

In the stochastic case, when the estimated value of the exponential parameter b_1 is significantly greater than 1, researchers have suggested that this is as an indicator of a diseconomy of scale. Correspondingly, when the estimate of b_1 is significantly less than 1, researchers have suggested that this is an indicator of economy of scale, see for example Banker et al. (1994), Hu (1997) and Kitchenham (2002). If this interpretation is robust towards change from one meaningful model of the size–effort relationship to another, we would expect the reverse model to behave in the same fashion as the deterministic model, i.e., if we find $b_1 > 1$ then using the same dataset we should find $b_2 < 1$, alternatively if find $b_1 < 1$ then using the same dataset we should find $b_2 > 1$. However, if we find that the estimates of the parameters do not behave in this fashion, i.e., in the same data set the estimates of b_1 and b_2 are both less than 1, or both greater than 1, such that the estimate of one parameter indicates an economy of scale but the other indicates a diseconomy of scale, we have an interpretation problem. We have then simultaneously found evidence for economy of scale and for diseconomy of scale on the same data set. Unless we have good reasons to trust one of the two models (one of the two regression lines) and not the other, all that we can conclude is that the data set cannot provide any reliable information about economies of scale.