

MIT LIBRARIES DUPL 1



3 9080 00562492 6

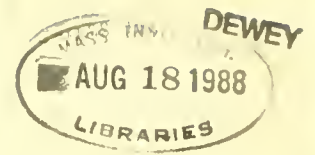
BASEMENT







3  
4  
995-  
88



**SCALE ECONOMIES  
IN NEW SOFTWARE DEVELOPMENT**

**Rajiv D. Banker  
Chris F. Kemerer**

**February 1988**

**CISR WP No. 167  
Sloan WP No. 1995-88**

**Center for Information Systems Research**

Massachusetts Institute of Technology  
Sloan School of Management  
77 Massachusetts Avenue  
Cambridge, Massachusetts, 02139



**SCALE ECONOMIES  
IN NEW SOFTWARE DEVELOPMENT**

**Rajiv D. Banker  
Chris F. Kemerer**

**February 1988**

**CISR WP No. 167  
Sloan WP No. 1995-88**

©1988 R.D. Banker, C.F. Kemerer

**Center for Information Systems Research  
Sloan School of Management  
Massachusetts Institute of Technology**

M.I.T. LIBRARIES  
AUG 1 1950  
RECEIVED



# SCALE ECONOMIES IN NEW SOFTWARE DEVELOPMENT

Rajiv D. Banker  
Professor of Management  
Carnegie Mellon University

Chris F. Kemerer  
Assistant Professor of Management Science  
Massachusetts Institute of Technology

## ABSTRACT

In this research we reconcile two opposing views regarding the presence of economies or diseconomies of scale in new software development. Our general approach hypothesizes a production function model of software development that allows for both increasing and decreasing returns to scale, and argues that *local* scale economies or diseconomies depend upon the size of projects. Using eight different data sets, including several reported in previous research on the subject, we provide empirical evidence in support of our hypothesis. Through use of the nonparametric DEA technique we also show how to identify the most productive scale size that may vary across organizations. These results are extended to include the effects of nonparametric scale-related factors, such as project duration and the number of new staff on the project team.



## 1. RESEARCH PROBLEM

Software development practitioners are faced with the problem of how to appropriately size new software development projects so as to maximize productivity. Unfortunately, much of the research in this area has arrived at apparently contradictory conclusions, namely that either economies of scale exist or that diseconomies of scale exist. This paper integrates these apparently contradictory results in a consistent framework, and empirically demonstrates that the existence of *local* scale economies or diseconomies depends upon the size of software development projects.<sup>1</sup> In addition, we provide a methodology for identifying the most productive scale size for a given software development environment, and show the effects of some other scale related variables on productivity.

A production process exhibits (local) increasing returns to scale if, at a given volume level, the marginal returns of an additional unit of input exceed the average returns. Economies of scale are thus present when average productivity is increasing, and scale diseconomies prevail when average productivity is decreasing. Reasons provided to explain the presence of economies of scale range from specialization of labor to phenomena such as learning curves. Software engineering researchers such as Boehm [16] have noted the presence of a number of factors in new software development that may contribute to economies of scale, such as software development tools like online debuggers or code generators. These tools may increase productivity, but the relatively large initial investment, both in purchase and in the organizational learning costs, may proscribe their use on small projects. Larger projects may also benefit from specialized personnel, whose expertise in a certain area (e.g., assembly language coding) may increase the project's overall productivity. Finally, all projects require a certain fixed investment in project management overhead. This type of overhead (e.g., status meetings and reports)

---

<sup>1</sup>In production economics, economies of scale (are) defined at specific volume levels in a production process, and are thus best described as local. It is therefore inappropriate to limit the characterization of a production process to only global economies (or diseconomies) of scale. In dealing with single input-single output production correspondences, we shall use the terms increasing returns to scale and scale economies interchangeably.

does not increase directly with project size and therefore can be a source of economies of scale for larger projects.

In contrast to the reasons cited above, many authors have pointed out the possibility of diseconomies of scale on large software projects. Brooks [17] has suggested that the number of communication paths between project team members increases (at an increasing rate) with the number of team members.<sup>2</sup> This communication overhead is a clear case of nonlinear cost increase, and hence a factor that could contribute to diseconomies of scale. Somewhat analogously, Conte et al. [21] suggest that larger systems development projects will face more complex interface problems between system components. Boehm [16] points out that increasing the number of people also increases the chances for personality conflicts among team members. Jones [25] notes that many overhead activities, such as planning and documentation, grow at a faster than linear rate as project size increases. Another possible source of diseconomies of scale is project slack, which is likely to be larger on a larger project and may contribute to reduced productivity.

Given these contradictory hypotheses, how can researchers best model the software development production process? And, how can practicing software development managers appropriately size new software development projects so as to maximize average productivity? This paper addresses these questions and is organized as follows. Section 2 presents the empirical evidence for both the notion of economies of scale and the notion of diseconomies of scale in new software development. We integrate these two notions and suggest that in most organizations, the software development production process first exhibits (local) increasing returns to scale, but decreasing returns set in for very large projects. We believe that one reason that this has not been shown by other researchers is due to the simple parametric models employed. We show in Section 3, however, that in empirical applications even the more flexible parametric forms are limited in

---

<sup>2</sup>The number of paths required is  $n(n-1)/2$ , where  $n$  is the number of project team members.

their ability to estimate the returns to scale. This motivates our use in Section 4 of Data Envelopment Analysis as an alternative nonparametric modeling technique to identify the most productive scale size. Section 5 presents the results of further analysis of other scale-related variables. Finally, the conclusions and suggestions for further research are presented in Section 6.

## 2. EMPIRICAL EVIDENCE

A number of researchers have collected empirical data that support increasing returns to scale theories. The general approach of these researchers has been to estimate a function of the form:

$$y = a(x)^b$$

where  $y$  is the amount of input, typically professional work-hours, and  $x$  is the size of the project, typically measured in terms of source lines of code (SLOC) or Function Points (FP). This function is estimated by taking the logarithms of both sides and then estimating the resulting linear model using regression techniques.

$$(1) \ln(y) = a + b \ln(x)$$

An estimated exponent value,  $b$ , less than 1 indicates economies of scale, while an exponent greater than 1 indicates diseconomies of scale. This follows because the returns to scale measure is

$$\rho = \frac{x}{y} \frac{dy}{dx} = b.$$

That is, marginal productivity ( $dx/dy$ ) is greater than (less than) average productivity ( $x/y$ ) if  $b$  is less than (greater than) one.

One of the earliest pieces of research to estimate this function was the work of Walston and Felix [31]. They collected data on 60 projects within IBM's Federal Systems Division and estimated a function with an exponent of .91, a result that would indicate mild increasing returns to scale. Jeffery and Lawrence [24] and Vessey [30] have also reported economies of scale on small projects, although they have not published their data.

We have extended this analysis to a number of other published data sets. Using the 1978-80 data from a Yourdon [22] survey of 17 projects from a

variety of firms, we estimated an exponent of .72, indicating increasing returns to scale. Two other data sets that display exponents of approximately .95 are from Bailey's study [3] of 19 NASA/Goddard Space Flight Center projects and Behren's study [14] of 25 projects at Equitable Life Assurance Society.<sup>3</sup> Kemerer's [26] Function Point data from a commercial data processing consulting firm yield an estimated exponent of .85. In summary, the evidence for economies of scale comes from a number of sources representing a wide variety of application environments.

However, a number of researchers have provided empirical support for the notion of diseconomies of scale as well. Boehm's [16] 63 project COCOMO data set exhibits an exponent of 1.11. We estimated a high exponent of 1.49 for Albrecht's [2] 24 projects from IBM measured in Function Points. Two data sets that produce identical exponents of 1.06, showing mild decreasing returns to scale are Belady and Lehman's [15] 33 project data set from a large software house and Wingfield's [32] 15 project U.S. Army data set. Therefore, the empirical evidence for diseconomies of scale in new software development is at least as compelling as that for economies of scale.

Table 1 summarizes the loglinear model analysis of the nine data sets, with five exhibiting increasing returns to scale and four exhibiting decreasing returns to scale. Note that the data sets in Table 1 are listed in the approximate order of their average size.<sup>4</sup> One interesting result available from even a cursory examination of Table 1 is that the data sets with smaller average projects (the first four on the list) tend to show economies of scale, while the data sets with larger average projects tend to show diseconomies of scale. The last data set, which contains the largest projects on the list, is the only exception to this simple analysis. This data set (also referred to as the ABC data set) is described in Table 2.

---

<sup>3</sup> Behren's data set is not reported directly in his paper. However, a scatter graph is provided, which was enlarged and the data directly extrapolated. Note that the graph contains only 22 of the 25 reported data-points.

<sup>4</sup> Behren's data is only provided in terms of Function Points. Using Albrecht's linear model, the average SLOC for Behren's data is estimated to be approximately 10.8K.

The returns to scale results reported in Table 1 thus indicate that the conflicting theories about the presence of scale economies or diseconomies described in Section 1 are matched by conflicting empirical evidence obtained for different data sets. We reconcile this apparent contradiction by offering the hypothesis that for most software development "production processes" there exist increasing returns to scale for smaller projects and decreasing returns for very large projects. That is, average productivity is increasing as long as the project size is smaller than the "most productive scale size" (MPSS), and is decreasing for projects that are larger.<sup>5</sup> The actual MPSS may be different for different organizational settings.

The reasons for our above hypothesis stem from the conflicting arguments presented earlier in Section 1 for the presence of both economies and diseconomies of scale. Since most projects require a significant fixed investment in project management overhead, average productivity increases initially as the fixed overhead is spread over a larger project. But the larger project size generally makes it more difficult to manage, and the marginal productivity of the project team is likely to decline. Increasing returns continue to prevail as long as average productivity remains less than marginal productivity. At the most productive scale size (MPSS) marginal productivity equals average productivity, and beyond MPSS average productivity, being greater than marginal productivity, is declining and decreasing returns to scale prevail. This intuitive argument is depicted in Figure 1. In the next two sections we reexamine eight<sup>6</sup> of the nine data sets within the framework of less restrictive estimation models to provide empirical support for our hypothesis.

The MPSS will tend to differ across organizations. If the fixed overhead is large, or if the marginal productivity does not decline rapidly, increasing returns will

---

<sup>5</sup> Benker [4] provides a rigorous definition and discussion of the concept of most productive scale size (MPSS). We pursue this analysis further in Section 4.

<sup>6</sup> Walston and Felix [31] report their estimated loglinear model, but do not present the actual data.

continue to prevail for larger projects and the MPSS will be large. On the other hand, if the fixed overhead is relatively small or if the marginal productivity declines sharply, then the MPSS is small and decreasing returns set in at a lower scale level.

### 3. PARAMETRIC PRODUCTION FUNCTION ANALYSIS

The problem with the simple loglinear model of the previous research is that it does not allow for the possibility of increasing returns for some projects and decreasing for others. The estimated returns to scale are determined by a single parameter, the exponent  $b$ . But we require a more general model that allows for average productivity increases as the fixed project overhead gets spread over larger and larger projects, and after reaching the most productive scale size (MPSS), it allows for declining average productivity caused by negative factors affecting large projects such as the proliferation of communication paths. Rather than reject the parametric approach based only on the simple loglinear model, we first explore more flexible parametric forms that have been employed in empirical research in other production environments. Such a model that estimates MPSS would also be of use to software development managers because they can then identify the scale size where average productivity is maximized in their organization.

One possible method for generalizing the restrictive loglinear production function for new software development of previous research is by simply adding a logquadratic term as an independent variable. We can thus estimate the following translog function:<sup>7</sup>

$$(2) \ln(HOURS) = \beta_0 + \beta_1(\ln(SIZE)) + \beta_2(\ln(SIZE))^2$$

Letting  $y$  equal HOURS and  $x$  equal SIZE, it is evident that the returns to scale measure  $\rho$  is given by

$$(3) \rho = \frac{d \ln y}{d \ln x} = \frac{x dy}{y dx} = \beta_1 + 2\beta_2(\ln x)$$

---

<sup>7</sup>Christensen, Jorgenson and Lau [20] note that the translog is a flexible functional form that provides a local second-order approximation to an arbitrary, twice-continuously-differentiable production function.



Therefore, if the estimated  $\beta_2 > 0$  then increasing returns to scale prevail for  $x < \exp\{(1-\beta_1)/2\beta_2\}$  and decreasing returns prevail for project sizes greater than the estimated MPSS given here by  $x^* = \exp\{(1-\beta_1)/2\beta_2\}$ . If, however, the estimated  $\beta_2 < 0$  then average productivity is estimated to be increasing for smaller projects, and decreasing for longer projects, contrary to our arguments presented earlier.

Hildenbrand [23], Varian [29] and Banker and Maindiratta [9] have argued that such a parametric approach imposes considerable untested structure on the production function. To provide evidence of robustness of their results, several empirical studies therefore estimate different parametrically specified functional forms. For instance, in our present context an alternative specification may be the following quadratic form:

$$(4) \text{ HOURS} = \beta_0 + \beta_1(\text{SIZE}) + \beta_2(\text{SIZE})^2$$

Again letting  $y$  equal HOURS and  $x$  equal SIZE, the returns to scale measure  $\rho$  is given by:

$$(5) \rho = \frac{x \, dy}{y \, dx} = \frac{x(\beta_1 + 2\beta_2 x)}{y} = \frac{\beta_1 x + 2\beta_2 x^2}{\beta_0 + \beta_1 x + \beta_2 x^2}$$

Therefore,  $\rho > 1$  if and only if  $\beta_2 x^2 > \beta_0$ . If the estimated values of both  $\beta_0$  and  $\beta_2$  are positive then the MPSS is given by  $x^* = \sqrt{\beta_0/\beta_2}$ , with increasing returns for  $x < x^*$  and decreasing returns for  $x > x^*$ . If estimated  $\beta_0 < 0$  and  $\beta_2 > 0$  then decreasing returns are exhibited for all  $x > 0$ , and if estimated  $\beta_0 > 0$  and  $\beta_2 < 0$  then increasing returns are exhibited for all  $x > 0$ . If the estimated values of both  $\beta_0$  and  $\beta_2$  are negative then decreasing returns correspond to small projects and increasing returns correspond to large projects, contrary to our earlier hypothesis.

The empirical results for the eight available data sets for the logquadratic (translog) and the quadratic models are presented in Tables 3 and 4 respectively.

Two empirical problems are encountered in practice. First, several researchers have observed that these so-called flexible parametric functional forms frequently violate reasonable regularity conditions, such as monotonicity, see for instance, Caves and Christensen [18] and Barnett and Lee [13]. In our present context, the logquadratic models estimated for the Bailey and Wingfield data sets exhibit  $dy/dx < 0$  (decreasing labor requirement for increasing project size) for smaller projects. Similar violation of the monotonicity condition is exhibited by the estimated quadratic models; for small projects by the Albrecht and Kemerer data sets and for large projects by the Bailey, COCOMO and Belady data sets.

The second empirical problem is more serious for our objective of estimating returns to scale for new software development. The pairs of independent variables  $\ln(SIZE)$  and  $(\ln(SIZE))^2$ , and  $(SIZE)$  and  $(SIZE)^2$ , tend to be highly correlated. The range of pairwise correlations was 0.967 - 0.999 for  $\ln(SIZE)$  and  $(\ln(SIZE))^2$  and 0.915 - 0.974 for  $(SIZE)$  and  $(SIZE)^2$  for the eight available data sets. This high level of collinearity implies that the confidence about interpreting the estimates of the coefficients  $\beta_1$  and  $\beta_2$  as the change in the dependent variable due to a change in the independent variables will be very low for both the logquadratic and the quadratic models.<sup>8</sup> Consequently, the estimates of these coefficients are likely to be unstable, see for instance Pindyck and Rubinfeld [27]. The usual econometric methods, therefore, may not be appropriate for estimating the nature of returns to scale or the most productive scale size for these eight data sets.<sup>9</sup>

The high collinearity between  $\ln(SIZE)$  and  $(\ln(SIZE))^2$  is also of importance to the interpretation of the results of the estimation of the simple loglinear models reported in Table 1. The estimated coefficient  $b$  in this case is likely to also pick up the effect of the omitted variable  $(\ln(SIZE))^2$ , and therefore, the interpretation of  $b$  as the estimated returns to scale measure may not be appropriate.

---

<sup>8</sup> The standard errors of the estimated coefficients are likely to be larger, and the corresponding t-statistics are less likely to be significant when the independent variables are highly correlated.

<sup>9</sup> The variance of the estimates of the returns to scale or MPSS measures depend on the variance and the covariance of the estimates of  $\beta_0$ ,  $\beta_1$  and  $\beta_2$ .

#### 4. NONPARAMETRIC PRODUCTION FUNCTION ANALYSIS

Given these problems, and the limited *a priori* knowledge about the functional form of the production process underlying software development, specifying a parametric form for the production correspondence is difficult to substantiate theoretically or validate statistically. Also, it is not immediately apparent what restrictions these hypotheses, treated as axioms in the econometric approach, impose on the production correspondence [9, 23, 29]. Production economics theory indicates the need to employ a frontier notion for a production function, with deviations from the frontier occurring due to inefficiencies exhibited in individual observations [8, 29]. This differentiates between characteristics of the process and individual inefficiencies. Therefore, we propose to use Data Envelopment Analysis (DEA), a nonparametric approach to production frontier estimation developed by Charnes, Cooper and Rhodes [19] and extended to a formal production economics framework by Banker, Charnes, and Cooper [5].<sup>10</sup> DEA does not impose a parametric form on the production function and assumes only that a monotonic and convex relationship exists between inputs and products.<sup>11</sup> More formally, the following limited assumptions are made about the frontier production function  $f(x)$  :

1. Monotonicity : If  $y = f(x)$ ,  $y' = f(x')$  and  $x \geq x'$ , then  $y \geq y'$
2. Convexity : If  $y = f(x)$ ,  $y' = f(x')$  and  $0 \leq \lambda \leq 1$  then  $(1-\lambda)y + \lambda y' \geq f[(1-\lambda)x + \lambda x']$
3. Envelopment : For each observation  $k$ ,  $k = 1, \dots, n$ ,  $y_k \geq f(x_k)$
4. Minimum Extrapolation : If a function  $g(\cdot)$  satisfies the monotonicity, convexity and envelopment conditions, then  $g(x) \leq f(x)$  for all  $x$ .

The estimation of the function  $f(x)$  can be accomplished using linear programming techniques, and estimates of  $f(x_j)$  obtained in this manner are maximum likelihood and consistent, see Banker [12]. The most productive scale size is

---

<sup>10</sup>Recent developments [10] in stochastic data envelopment analysis simultaneously consider deviations from the production frontier due to inefficiencies and also measurement errors.

<sup>11</sup>Evidence on the comparative application of the DEA and translog models is provided by Banker, Conrad and Strauss [6].

estimated via the following linear programming model as in Banker [4] for the general case of multiple inputs and outputs.<sup>12</sup>

$$(6) \min \eta_A$$

subject to

$$(6.1) \sum_{k=1}^n \lambda_k x_{jk} \geq x_{jA} \quad j=1, \dots, J$$

$$(6.2) \sum_{k=1}^n \lambda_k y_{ik} \leq \eta_A y_{iA} \quad i=1, \dots, I$$

$$(6.3) \eta_A, \lambda_k \geq 0$$

where  $x_{jk}$  = output  $j$ , for observation  $k$ ,

$y_{ik}$  = input  $i$ , for observation  $k$ , and

$n$  = number of observations ( $k = 1, 2, \dots, n$ ).

The MPSS for the input-output mix given by  $(y_A, x_A)$  where  $y_A \equiv (y_{1A}, \dots, y_{iA}, \dots, y_{IA})$ , and  $x_A \equiv (x_{1A}, \dots, x_{jA}, \dots, x_{JA})$  is computed as follows:

$$MPSS = \frac{x_A}{\eta_A \sum_{k=1}^n \lambda_k}$$

In our present context, we are interested only in a single input-single output production correspondence, and the computational problem is consequently considerably simplified. The solution to the linear program in (6) is given by simply  $\eta_A^* = x_A / y_A M$  where  $M = \max_k \{x_k / y_k \mid k=1, \dots, n\}$  is the maximum observed

---

<sup>12</sup> Alternative models for estimating MPSS when some inputs or outputs are fixed or uncontrollable, or when some variables are measured on a categorical rather than on a continuous scale are described by Banker and Morey (7). Banker and Maindiratta (8) discuss the estimation of other non-convex technologies.

average productivity across all observations. The most productive scale size is given by the project size  $x_M$ , say, for which  $x_M/y_M (=M)$  is the largest for all observations. If the maximum average productivity is also attained for some other observation, say  $M'$ , with observed input-output pair  $(x_{M'}, y_{M'})$ , then there is evidence of local constant returns to scale, and the range of project sizes between  $x_{M'}$  and  $x_M$  all represent MPSS.

The MPSS was calculated for the eight available data sets, and the results are reported in Table 5 using the size metric chosen by each researcher. From a practitioner's viewpoint, the MPSS provides a project size goal in order to maximize the average productivity of future new software development projects. From a research perspective, it also allows the identification of both increasing and decreasing returns within these empirical data sets. Projects larger (smaller) than the MPSS correspond to decreasing (increasing) returns, respectively. Table 5 shows the MPSS and the corresponding percentile value for the range of observed output data for each of the eight data sets. In five of the eight cases, the MPSS is within the inter-quartile range for the observed output data, thus indicating that both increasing and decreasing returns are clearly present since there exist both smaller and larger projects than the MPSS at that site. It follows therefore that the loglinear model may be an inadequate description of many new software development application environments.

## 5. OTHER SCALE-RELATED FACTORS

In our analysis in the earlier sections, we have considered only the project size as measured by SLOC or Function Points as the factors explaining the variations in the professional labor requirements of different projects. Other scale-related factors, such as the duration of a project, or the number of new staff members assigned to a project team may also explain these variations. While the methodology employed for our earlier analysis remains applicable and provides an important link to prior research, the specific results must be interpreted cautiously to the extent that such additional factors have not been included in the analysis. In

this section, we consider the explicit inclusion of other scale-related factors in our estimation models, and we provide empirical evidence about the importance of these factors.

In addition to size as measured by SLOC or Function Points, project scale is often represented by the amount of calendar time taken by a project. A project's duration, while possibly highly correlated with a size measure, need not be. For example, a project containing a given number of SLOC could be stretched out over a relatively longer period of time. For a project composed of a number of tasks this might be due to a situation where one task was required to be completed before the others could commence due to their logical dependence. An example might be getting a critical user-signoff on some piece of the project. This would result in a number of other tasks that are not on the "critical path" having a lot of slack. Abdel-Hamid and Madnick [1] suggest that this type of effect also occurs on projects that have mis-estimated the amount of time required.

Increasing the duration on a project is likely to increase the number of work-hours expended on the project for a couple of reasons. Over the course of a long project there is more chance of changes in user requirements, thus necessitating re-work. Also, Boehm [16] suggests that with the likely additional slack on a longer project a "Parkinson's Law" type ("work expanding to fill the time available for its completion") effect occurs.

In order to test the significance of the project's calendar time as a scale-related explanatory variable, regression analysis was performed on the six data sets where the calendar time (DURATION) was available.<sup>13</sup> The regression models were of the form:

$$(7) \text{ HOURS} = \beta_0 + \beta_1(\text{SIZE}) + \beta_2(\text{SIZE})^2 + \beta_3(\text{DURATION})$$

This model was chosen to represent the nonlinear aspects of the HOURS : SIZE relationship. The variables SIZE and (SIZE)<sup>2</sup> are highly correlated as noted

---

<sup>13</sup>For the Bailey, Yourdon, Belady, and Wingfield data sets, this data is available in Conte [21].

earlier, and therefore the estimation of the corresponding parameters will suffer from collinearity problems. However, this does not affect the objective of our analysis which is to identify whether DURATION provides a significant explanation of the variation in HOURS, over and above that provided by the SIZE and (SIZE)<sup>2</sup> variables. The results of this analysis are shown in Table 6. Of the six data sets, DURATION is statistically significant at the 95% confidence level in four. Of the other two, in the Yourdon data DURATION is highly correlated with SIZE (correlation coefficient = .71, see Table 7) which provides a possible reason why it is not adding any additional explanatory power. Only in the Wingfield data does the DURATION variable not seem to add to the labor requirement.

Another scale-related factor affecting productivity is suggested by the literature dealing with the communication paths on a project. Not only could additional overhead be related to the *number* of paths, but also to the *efficiency* of those paths. This could be affected by how much experience the project team had working as a group. A variable, NEWSTAFF, the number of new staff members on a project team, has been suggested by Rubin [28] as a possible source of productivity variation. While a larger project is likely to require more people and be assigned more new staff, this need not be the case. The NEWSTAFF data are currently available only for the Kemerer data set [26]. To identify the impact of this additional factor we estimate the following model:

$$(8) \text{ HOURS} = \beta_0 + \beta_1(\text{SIZE}) + \beta_2(\text{SIZE})^2 + \beta_3(\text{DURATION}) + \beta_4(\text{NEWSTAFF})$$

The results for the ABC data are:

$$\text{Hours} = 16042 - 93.93(\text{FP}) + 0.06(\text{FP})^2 + 1522(\text{DURATION}) + 9.06(\text{NEWSTAFF})$$

(1.43)    (-4.10)            (6.62)            (2.71)                    (2.10)

The t-statistics are reported in parentheses.

$$R^2 = .90$$

The correlation matrix for these variables is:

NEWSTAFF    DURATION    FP

DURATION	-0.104		
FP	-0.165	0.338	
(FP) <sup>2</sup>	-0.221	0.203	0.950

From these results, it is apparent that the number of new staff members is a significant predictor of the variation in work-months for the Kemerer data set, over and above the other scale-related variables Function Points, (Function Points)<sup>2</sup> and Duration. It is important to note that this is the number of *new* staff members, not simply the number of staff. For this data set, the correlation of NEWSTAFF and average staff (the ratio of work-months to calendar months) is only .02.

An alternative nonparametric estimation of this production function can be accomplished by using an extension of DEA, see Banker [10]. The following general model is specified:

$$(9) \text{ HOURS} = f(\text{FP}) + \beta_1(\text{DURATION}) + \beta_2(\text{NEWSTAFF})$$

Rather than assuming a specific parametric functional form (such as the quadratic), in this DEA model the function  $f(\text{FP})$  is assumed only to be monotone increasing and convex, while DURATION and NEWSTAFF represent additive linear terms. The results for this case are coefficients of 1018 for DURATION and 1163 for NEWSTAFF. The deletion of the variable NEWSTAFF from the model in (9) increases the sum of squared residuals by 123.1% and the sum of absolute residuals by 88.9%. Similarly, when we remove the variable DURATION from the model in (9), the sum of squared residuals increases by 109.9% and the sum of absolute residuals increases by 76.6%. Thus, each of these two variables provides substantial incremental explanatory power to the model, see Banker and Morey [11]. These results are not dependent upon pre-selecting a functional form but are generally consistent with the results from the quadratic model, thus corroborating the conclusions drawn from our earlier regression analysis.



## 6. CONCLUDING REMARKS

In this research we have reconciled two opposing views regarding the presence of economies or diseconomies of scale in new software development. Our general approach provides for a production function model of software development that allows for both increasing and decreasing returns to scale. Through use of the DEA technique we have also shown how to identify the most productive scale size. These results were extended to include the effects of other scale-related factors.

For the practitioner, our results contain a number of useful implications. In terms of project estimation, traditional algorithmic models have suggested a simple loglinear model with which to estimate eventual work-hours. While these models have some limited applicability, they ignore the possibility of improving project productivity by carefully selecting the scale of the project. Rather than taking the scale as exogeneous, as most of these simple models do, managers could actively seek to identify the most productive scale size for their organization. Our results suggest that this MPSS varies widely across different application environments, and an interesting extension to this work would be to identify factors that contribute to some organizations' ability to successfully manage larger projects. Managers could also assess the effects on productivity of other scale-related factors, such as calendar duration and the number of new project team staff members.

Table 1: Summary of Loglinear Models

DATA SET	n	MEAN SLOC	MEAN FP	b	t-statistic $H_0: b=1$
Behrens	22	n.a.	146	.94	-.32
Walston	60	20K	n.a.	.91	n.a.
Bailey	19	29K	n.a.	.95	-.73
Yourdon	17	34K	n.a.	.72	-1.21
COCOMO	63	67K	n.a.	1.11	1.30**
Albrecht	24	66K	648	1.49	2.57*
Belady	33	92K	n.a.	1.06	.60
Wingfield	15	180K	n.a.	1.06	.20
Kemerer	17	220K	1013	.85	-.79

\*Significant at the 5 percent level for a one-tailed test.

\*\*Significant at the 10 percent level for a one-tailed test.

Table 2: Kemerer [26] Data Set

Project	Hours	Func Pts	KSLOC	Duration	Newstaff
1	43624	1217.1	253.6	17	0
2	12540	507.3	40.5	7	8
3	168311	2306.8	450.0	15	0
4	13209	788.5	214.4	18	4
5	51118	1337.6	449.9	13	35
6	12768	421.3	50.0	5	15
7	3526	99.9	43.0	5	4
8	19806	993.0	200.0	11	12
9	17632	1592.9	289.0	14	2
10	10944	240.0	39.0	5	7
11	39322	1611.0	254.2	13	3
12	35066	789.0	128.6	31	10
13	23864	690.9	161.4	20	10
14	37529	1347.5	164.8	26	3
15	10625	1044.3	60.2	14	0
16	11552	1209.0	155.0	9	5
17	15048	1030.0	195.0	13	8

Table 3: Summary of Logquadratic Models

DATA SET	$\beta_0$	$\beta_1$	$\beta_2$	R <sup>2</sup>	MPSS*
Bailey	6.27 (13.82)	-.54 (1.48)	.075 (1.14)	92.5%	21.47 KSLOC
Yourdon	7.04 (2.62)	.17 (.10)	.08 (.33)	39.8%	148.41 KSLOC
COCOMO	6.56 (12.10)	.985 (2.92)	.05 (.38)	73.8%	1.16 AKDSI
Belady	6.08 (5.59)	1.39 (2.31)	-.04 (-.53)	78.3%	130.97 KSLOC
Wingfield	20.90 (1.75)	-4.68 (-.97)	.57 (1.20)	55.4%	5.06 KSLOC
Behrens	6.57 (1.61)	-.91 (-.49)	.21 (1.00)	57.7%	98.63 FP
Albrecht	3.60 (.33)	.37 (.11)	.09 (.32)	73.6%	35.86 FP
Kemerer	11.50 (1.55)	-1.55 (-.64)	.19 (.99)	60.2%	739.68 FP

MPSS\*  $\equiv$  estimated most productive scale size.

The t-statistics are presented in parentheses.

KSLOC  $\equiv$  thousands of source lines of code.

ADKSI  $\equiv$  thousands of adjusted delivered source instructions

FP  $\equiv$  function points

Table 4: Summary of Quadratic models

DATA SET	$\beta_0$	$\beta_1$	$\beta_2$	R <sup>2</sup>	MPSS*
Bailey	-1728 (-1.36)	517 (6.07)	-3.10 (-3.07)	88.6%	23.61 KSLOC
Yourdon	241 (.07)	261 (1.74)	-.90 (-.81)	42.8%	$\infty$
COCOMO	-31601 (-1.09)	2576 (6.28)	-1.51 (-3.12)	57.6%	144.66 AKDSI
Belady	-49442 (-.84)	2811 (3.17)	-2.50 (-1.76)	46.1%	140.74 KSLOC
Wingfield	81627 (.62)	13 (.01)	3.72 (1.31)	71.4%	148.13 KSLOC
Behrens	322 (.44)	4 (.39)	.029 (1.06)	58.4%	105.74 FP
Albrecht	8478 (1.87)	-14 (-1.09)	.034 (5.59)	94.9%	499.41 FP
Kemerer	26268 (1.95)	-59 (-2.35)	.049 (4.60)	81.8%	734.43 FP

MPSS\*  $\equiv$  estimated most productive scale size.

The t-statistics are presented in parentheses.

Table 5: Most Productive Scale Sizes Using DEA

DATA SET	MPSS*	Percentile
Behrens (1983)	170.1 FP	68.2%
Bailey	7.8 KSLOC	26.3%
Yourdon	55.8 KSLOC	88.2%
COCOMO	15.0 Adjusted AKDSI	39.7%
Albrecht	199.1 FP	4.2%
Belady	35.1 KSLOC	66.7%
Wingfield	250.1 KSLOC	80.0%
Kemerer	1209.3 FP	64.7%

MPSS\*  $\equiv$  estimated most productive scale size.

Table 6: Duration Regression Results

DATA SET	R <sup>2</sup>	$\beta_3$ COEFFICIENT	DURATION t-STATISTIC
Bailey	90.9	321	1.97*
Yourdon	44.6	186	.64
COCOMO	67.2	9442	4.17*
Belady	55.3	7141	2.45*
Wingfield	71.6	1243	.26
Kemerer	86.6	1344	2.15*

\*Significant at the 5 percent level for a one-tailed test.

Table 7: Duration Regression: Correlation Between Independent Variables

DATA SET		CORRELATIONS		
		Hours	Size	(Size) <sup>2</sup>
Bailey	Size	.91		
	(Size) <sup>2</sup>	.79	.96	
	Duration	.60	.46	.38
Yourdon	Size	.63		
	(Size) <sup>2</sup>	.55	.95	
	Duration	.57	.71	.63
COCOMO	Size	.71		
	(Size) <sup>2</sup>	.55	.92	
	Duration	.74	.58	.33
Belady	Size	.64		
	(Size) <sup>2</sup>	.53	.95	
	Duration	.63	.46	.31
Wingfield	Size	.82		
	(Size) <sup>2</sup>	.85	.97	
	Duration	.40	.43	.42
Kemerer	Size	.74		
	(Size) <sup>2</sup>	.86	.95	
	Duration	.24	.34	.20



## References

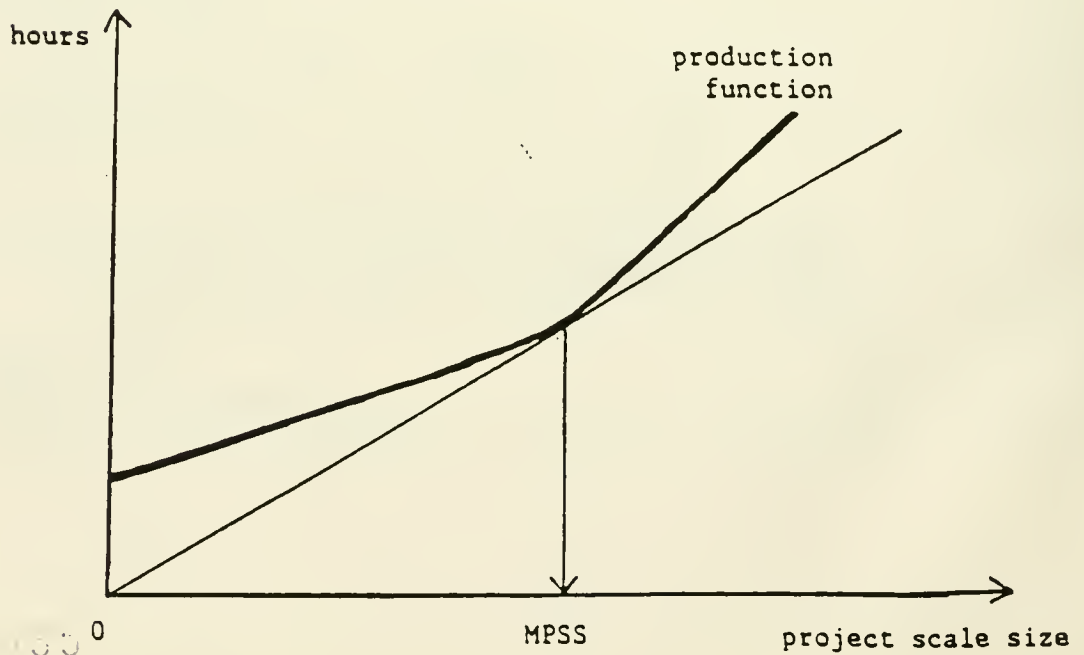
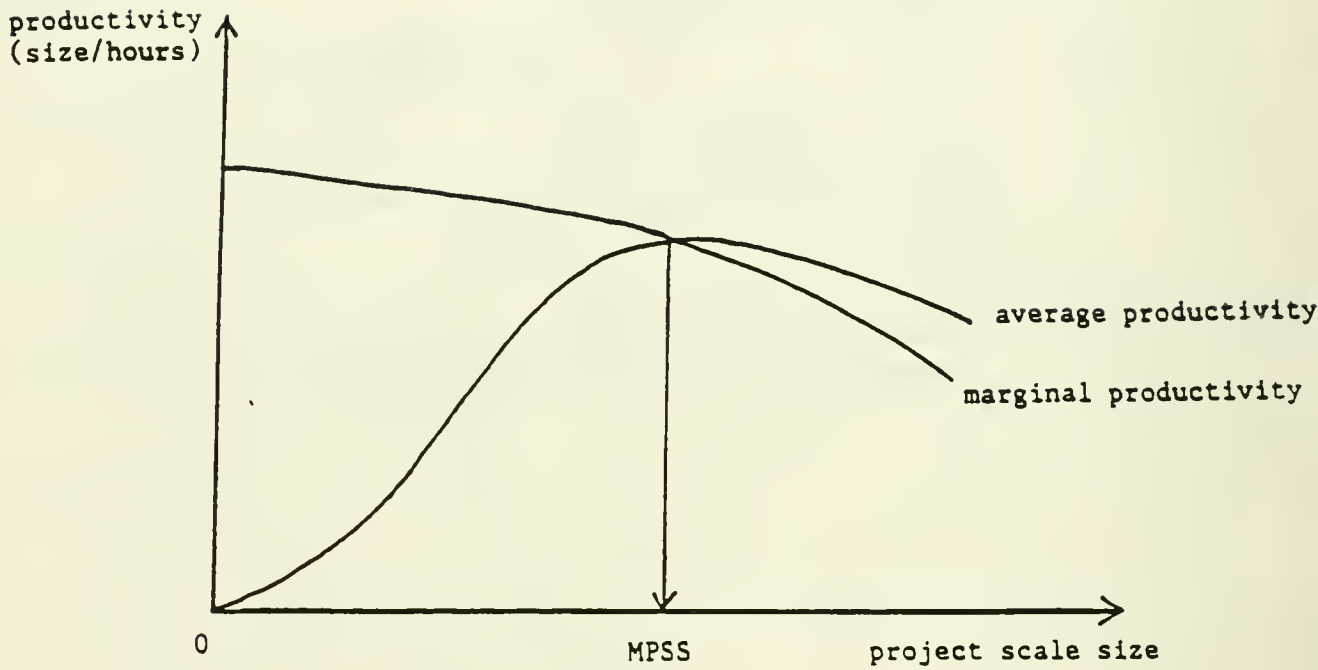
- [ 1 ] Abdel-Hamid, Tarek and Stuart Madnick.  
Impact of Schedule Estimation on Software Project Behavior.  
*IEEE Software* 3:70-75, July, 1986.
- [ 2 ] Albrecht, Allan J. and John Gaffney, Jr.  
Software Function, Source Lines of Code, and Development Effort  
Prediction: A Software Science Validation.  
*IEEE Transactions on Software Engineering* SE-9(6):639-648, November,  
1983.
- [ 3 ] Bailey, John W. and Victor R. Basili.  
A Meta-model for Software Development Resource Expenditures.  
In *Proceedings of the 5th International Conference on Software  
Engineering*, pages 107-116., 1981.
- [ 4 ] Banker, Rajiv.  
Estimating Most Productive Scale Size Using Data Envelopment Analysis.  
*European Journal of Operations Research* 17(1):35-44, July, 1984.
- [ 5 ] Banker, R. D., A. Charnes and W. W. Cooper.  
Some Models for Estimating Technical and Scale Inefficiencies in DEA.  
*Management Science* 30(9):1078-1092, September, 1984.
- [ 6 ] Banker, R. D., R. F. Conrad and R. P. Strauss.  
A Comparative Application of DEA and Translog Methods: An Illustrative  
Study of Hospital Production.  
*Management Science* 32(1):30-44, January, 1986.
- [ 7 ] Banker, R. D. and R. C. Morey.  
Efficiency Analysis for Exogenously Fixed Inputs and Outputs.  
*Operations Research* 34(4):513-521, July-August, 1986.
- [ 8 ] Banker, R. D. and A. Maindiratta.  
Piecewise Loglinear Estimation of Efficient Production Surfaces.  
*Management Science* 32(1):126-135, January, 1986.
- [ 9 ] Banker, R. D. and A. Maindiratta.  
Nonparametric Analysis of Technical and Allocative Efficiencies in Production.  
1987.  
Forthcoming in *Econometrica*.
- [ 10 ] Banker, Rajiv.  
Stochastic Data Envelopment Analysis.  
1987.  
Carnegie Mellon University Working Paper, 1987.
- [ 11 ] Banker, R. D. and R. C. Morey.  
Evaluating Hypotheses in Efficiency Analysis: An Application.  
1987.  
Working Paper.
- [ 12 ] Banker, Rajiv.  
Maximum Likelihood, Consistency and Data Envelopment Analysis.  
Carnegie Mellon University Working Paper, 1987.

- [ 13] Barnett, W. A. and Y. W. Lee.  
The Global Properties of the Minflex Laurent, Generalized Leontief and  
Translog Flexible Functional Forms.  
*Econometrica* --:1421-1437, 1985.
- [ 14] Behrens, Charles A.  
Measuring the Productivity of Computer Systems Development Activities with  
Function Points.  
*IEEE Transactions on Software Engineering* SE-9(6):648-652, November,  
1983.
- [ 15] Belady, L. A. and M. M. Lehman.  
The Characteristics of Large Systems.  
In Wegner, P. (editor), *Research Directions in Software Technology*, pages  
106-138. MIT Press, Cambridge, MA, 1979.
- [ 16] Boehm, Barry W.  
*Software Engineering Economics*.  
Prentice-Hall, Englewood Cliffs, NJ, 1981.
- [ 17] Brooks, Frederick P.  
*The Mythical Man-Month*.  
Addison-Wesley, Reading, Mass., 1975.
- [ 18] Caves, D. W. and L. R. Christensen.  
Global Properties of Flexible Functional Forms.  
*American Economic Review* --:422-432, 1980.
- [ 19] Charnes, A., W. W. Cooper and E. Rhodes.  
Evaluating Program and Managerial Efficiency: An Application of Data  
Envelopment Analysis to Program Follow Through.  
*Management Science* 27(6):668-697, June, 1981.
- [ 20] Christensen, L. R., D. W. Jorgenson and L. J. Lau.  
Transcendental Logarithmic Production Frontiers.  
*Review of Economics and Statistics* 55:28-45, 1973.
- [ 21] Conte, S., H. Dunsmore and V. Shen.  
*Software Engineering Metrics and Models*.  
Benjamin/Cummings, Reading, MA, 1986.
- [ 22] DeMarco, Tom.  
*Yourdon Project Survey: Final Report*.  
Technical Report, Yourdon, Inc., September, 1981.
- [ 23] Hildenbrand, Werner.  
Short-Run Production Functions based on Microdata.  
*Econometrica* 49(5):1095-1125, September, 1981.
- [ 24] Jeffery, D.R. and M. J. Lawrence.  
An Inter-Organisational Comparison of Programming Productivity.  
In *Proceedings of the 4th International Conference on Software  
Engineering*, pages 369-377. 1979.
- [ 25] Jones, Capers.  
*Programming Productivity*.  
McGraw-Hill, New York, 1986.

- 23
- [26] Kemerer, Chris F.  
An Empirical Validation of Software Cost Estimation Models.  
*Communications of the ACM* 30(5):416-429, May, 1987.
  - [27] Pindyck, R.S. and D.L. Rubinfeld.  
*Econometric Models and Economic Forecasts*.  
McGraw-Hill, New York, 1981.
  - [28] Rubin, Howard A.  
Macroestimation of Software Development Parameters: The Estimacs System.  
In *SOFTFAIR Conference on Software Development Tools, Techniques and Alternatives*. IEEE, 1983.
  - [29] Varian, H.  
The Nonparametric Approach to Production Analysis.  
*Econometrica* 52(3):579-597, 1984.
  - [30] Vessey, Iris.  
On Program Development Effort and Productivity.  
*Information & Management* 10:255-266, 1986.
  - [31] Walston, C.E. and C.P. Felix.  
A Method of Programming Measurement and Estimation.  
*IBM Systems Journal* 16(1):54-73, 1977.
  - [32] Wingfield, C. G.  
*USACSC experience with SLIM*.  
Technical Report IAWAR 360-5, U.S. Army Institute for Research in  
Management Information and Computer Science, 1982.

Figure 1

Most Productive Scale Size (MPSS)



MIT LIBRARIES DUPL 2



3 9080 00562492 6





3-16-69

**Date Due**

Out to [unclear]

MAY 24 1991

FEB. 13 1993



BARCODE  
ON NEXT  
TO LAST  
PAGE

