Scale Economies in New Software Development

RAJIV D. BANKER AND CHRIS F. KEMERER

***Abstract*—In this paper 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 the use of the nonparametric DEA technique we also show how to identify the most productive scale size that may vary across organizations.**

***Index Terms*—Data envelopment analysis, function points, productivity measurement, scale economies, software development, source lines of code.**

I. 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.[1] In addition, we provide a methodology for identifying the most productive scale size for a given software development environment.

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 [14] 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) 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 [15] has suggested that the number of communication paths between project team members increases (at an increasing rate) with the number of team members.[2] 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.* [19] suggest that larger systems development projects will face more complex interface problems between system components. Boehm [14] points out that increasing the number of people also increases the chances for personality conflicts among team members. Jones [23] 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 II 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,

---

Manuscript received November 10, 1987; revised March 30, 1989. Recommended by W. Royce. R. Banker was supported in part by the National Science Foundation under Grant SES-8709044 at Carnegie-Mellon University.

R. Banker is with the Carlson School of Management, University of Minnesota, Minneapolis, MN 55455.

C. Kemerer is with the Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA 02139.

IEEE Log Number 8930142.

[1] 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.

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

0098-5589/89/1000-1199$01.00 © 1989 IEEE

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 III, however, that in empirical applications even the more flexible parametric forms are limited in their ability to estimate the returns to scale. This motivates our use in Section IV of Data Envelopment Analysis as an alternative nonparametric modeling technique to identify the most productive scale size. Finally, the conclusions and suggestions for further research are presented in Section V.

II. 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.

$$\ln(y) = a + b \ln(x). \tag{1}$$

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 the *reciprocal* of ρ where

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

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 [28]. They collected data on 60 projects within IBM's Federal Systems Division and estimated a function with an exponent of 0.91, a result that would indicate mild increasing returns to scale. Jeffery and Lawrence [22] and Vessey [27] 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 1978–1980 data from a Yourdon [20] survey of 17 projects from a variety of firms, we estimated an exponent of 0.72, indicating increasing returns to scale. Two other data sets that display exponents of approximately 0.95 are from Bailey and Basili's study [2] of 19 NASA/Goddard Space Flight Center projects and Behren's study [12] of 25 projects at Equitable Life Assurance Society.[3] Kemerer's [25] Function Point data from a commercial data processing consulting firm yield an estimated exponent of 0.85 (see Table I). 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 [14] 63 project COCOMO data set exhibits an exponent of 1.11. We estimated a high exponent of 1.49 for Albrecht and Gaffney's [1] 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 [13] 33 project data set from a large software house and Wingfield's [29] 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 I summarizes the loglinear model analysis of the nine data sets, with five exhibiting increasing returns to scale and four exhibiting decreasing returns to scale. The returns to scale results reported in Table II thus indicate that the conflicting theories about the presence of scale economies or diseconomies described in Section I 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.[4] 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 I 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. Productivity increases on progressively larger projects may also come from the greater use of specialized personnel and tools, and possibly greater management attention. But, eventually 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 marginal productivity remains greater than average 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 Fig. 1. We do not expect to observe situations where decreasing returns to scale prevail for smaller projects and increasing returns for larger projects. In the next two sections we reexamine

---

[3] 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.

[4] Banker [3] provides a rigorous definition and discussion of the concept of most productive scale size (MPSS). We pursue this analysis further in Section IV.

TABLE I
KEMERER [25] DATA SET

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

TABLE II
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.

eight[5] 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 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.

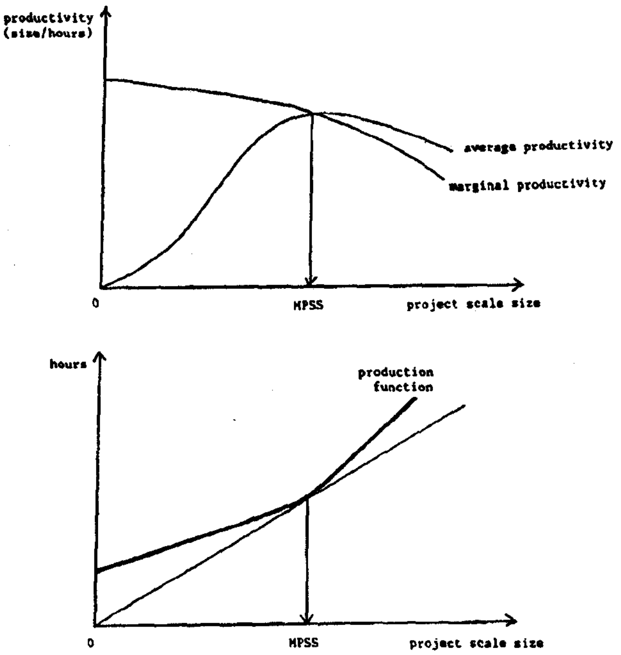

Fig. 1. Most productive scale size (MPSS).

III. 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 parameteric 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:[6]

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

---

[5] Walston and Felix [28] report their estimated loglinear model, but do not present the actual data.

[6] Christensen, Jorgenson, and Lau [18] note that the translog is a flexible functional form that provides a local second-order approximation to an arbitrary, twice-continuously-differentiable production function.

Letting y equal HOURS and x equal SIZE, it is evident that the reciprocal of the returns to scale measure is given by ρ where

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

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 β_2 were negative then it would be inconsistent with our arguments above that the average productivity is decreasing for smaller projects, and increasing for larger projects.[7]

Hildenbrand [21], Varian [26], and Banker and Maindiratta [8] 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.[8] For instance, in our present context an alternative specification may be the following quadratic form:

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

Again letting y equal HOURS and x equal SIZE, the reciprocal of the returns to scale measure is given by ρ where:

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

Therefore, $\rho > 1$ if and only if $\beta_2 x^2 > \beta_0$. If the estimated values of both β_0 and β_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 β_0 and β_2 are negative then decreasing returns correspond to small projects and increasing returns correspond to large projects, contrary to our earlier hypothesis.[9]

The empirical results for the eight available data sets for the logquadratic (translog) and the quadratic models are presented in Tables III and IV, 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 a monotonically increasing relation between inputs and outputs. See for instance, Caves and Christensen [16] and Barnett and Lee [11]. 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 this regularity 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.

TABLE III
SUMMARY OF LOGQUADRATIC MODELS

DATA SET	β_0	β_1	β_2	R^2	MPSS*
Behrens	6.57 (1.61)	-.91 (-.49)	.21 (1.00)	57.7%	98.63 FP
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
Albrecht	3.60 (.33)	.37 (.11)	.09 (.32)	73.6%	35.86 FP
Belady	6.06 (5.59)	1.39 (2.31)	-.04 (-.53)	76.3%	N/A
Wingfield	20.90 (1.75)	-4.68 (-.97)	.57 (1.20)	55.4%	5.06 KSLOC
Kemerer	11.50 (1.55)	-1.55 (-.64)	.19 (.99)	60.2%	739.68 FP

MPSS* ≡ estimated most productive scale size.

The t-statistics are presented in parentheses.

KSLOC ≡ thousands of source lines of code.

ADKSI ≡ thousands of adjusted delivered source instructions

FP ≡ function points

TABLE IV
SUMMARY OF QUADRATIC MODELS

DATA SET	β_0	β_1	β_2	R^2	MPSS*
Behrens	322 (.44)	4 (.39)	.029 (1.06)	58.4%	105.74 FP
Bailey	-1728 (-1.36)	517 (6.07)	-3.10 (-3.07)	88.6%	N/A
Yourdon	241 (.07)	261 (1.74)	-.90 (-.81)	42.8%	Increasing returns to scale for all observations
COCOMO	-31601 (-1.09)	2576 (6.28)	-1.51 (-3.12)	57.6%	N/A
Albrecht	8478 (1.87)	-14 (-1.09)	.034 (5.59)	94.9%	499.41 FP
Belady	-49442 (-.84)	2811 (3.17)	-2.50 (-1.76)	46.1%	N/A
Wingfield	61627 (.62)	13 (.01)	3.72 (1.31)	71.4%	148.13 KSLOC
Kemerer	26268 (1.95)	-59 (-2.35)	.049 (4.60)	81.6%	734.43 FP

---

[7] Such contrary evidence appears in one of the eight data sets examined by us. See Table III. The validity of this inference is questionable, however, because of the high (0.9 · · ·) correlation between the two independent variables in the regression.

[8] We assume here that software development effort is determined primarily by the size metric. If other factors are also considered relevant, they can be included in the estimation model and the MPSS estimated based on the corresponding coefficient estimates.

[9] Negative estimates for both β_0 and β_2 occur in three of the eight data sets. See Table IV. High correlation between the two variables (SIZE) and $(\text{SIZE})^2$ suggests that returns to scale estimates are likely to be unstable. In fact, in all these three contrary cases, the estimated coefficients indicate that less hours are required for larger projects.

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 β_1 and β_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.[10] Consequently, the estimates of these coefficients are likely to be unstable. See for instance Judge *et al.* [24]. 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.[11]

The high correlation 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 I. 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.

IV. 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 [8], [21], [26]. 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], [9]. 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 [17] and extended to a formal production economics framework by Banker, Charnes, and Cooper [4]. DEA does not impose a parametric form on the production function and assumes only that a monotonically increasing and convex[12] relationship exists between inputs and outputs, standard economic production function assumptions.[13] 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 $x \geq x'$, and $y \geq y'$.

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, $k = 1, \cdots, n$, $y_k \geq f(x_k)$.

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 [9]. The most productive scale size is estimated via the following linear programming model as in Banker [3] for the general case of multiple inputs and outputs.[14]

$$\min \eta_A \tag{6}$$

subject to

$$\sum_{k=1}^{n} \lambda_k x_{jk} \geq x_{jA} \qquad j = 1, \cdots, J \tag{6.1}$$

$$\sum_{k=1}^{n} \lambda_k y_{ik} \leq \eta_A y_{iA} \qquad i = 1, \cdots, I \tag{6.2}$$

$$\eta_A, \lambda_k \geq 0 \tag{6.3}$$

where

x_{jk} = output j for observation k,

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

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

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

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

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 $\eta_A^* = x_A / M y_A$ where $M = \max_k \{x_k / y_k \mid k = 1, \cdots, n\}$ is the maximum observed average productivity

---

[10] 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.

[11] The variance of the estimates of the returns to scale or MPSS measures depend on the variance and the covariance of the estimates of β_0, β_1, and β_2.

[12] The convexity assumption ensures that marginal productivity is decreasing so that decreasing returns for smaller projects are not followed by increasing returns for larger projects.

[13] Recent developments [10] in stochastic data envelopment analysis simultaneously consider deviations from the production frontier due to inefficiencies and also measurement errors. Evidence on the comparative application of the DEA and translog models is provided by Banker, Conrad, and Strauss [5].

[14] 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 [6]. Banker and Maindiratta [7] discuss the estimation of other nonconvex technologies.

TABLE V
MOST PRODUCTIVE SCALE SIZES USING DEA

DATA SET	MPSS*	Percentile	Inter-quartile Range
Behrens	170.1 FP	68.2%	Yes
Bailey	7.8 KSLOC	26.3%	Yes
Yourdon	55.8 KSLOC	88.2%	No
COCOMO	15.0 Adjusted AKDSI	39.7%	Yes
Albrecht	199.1 FP	4.2%	No
Belady	35.1 KSLOC	66.7%	Yes
Wingfield	250.1 KSLOC	80.0%	No
Kemerer	1209.3 FP	64.7%	Yes

Percentile ≡ percentage of projects less than or equal to the MPSS*.

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 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 V 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 V 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 interquartile range for the observed output data, thus indicating that both increasing and decreasing returns are 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.

V. CONCLUDING REMARKS

In this paper 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.

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. In order to estimate models for their own environments, managers will need to collect input and output metrics for their own projects. As we have demonstrated the general model with both SLOC and Functions Points, the choice of particular metrics can be made by the individual manager. The only critical consideration is that these data be collected consistently and accurately.

Another application of the MPSS idea is that once managers have estimated the MPSS for their organizations' software development process, this information could be used as input to the make or buy decision. If a new system were estimated to be of a size very different from the MPSS, then that would be an additional factor to take into account in favor of buying the system rather than developing it in-house.

Our results suggest that the 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.[15]

REFERENCES

[1] J. Albrecht and J. G. Gaffney, Jr., "Software function, source lines of code, and development effort prediction: A software science validation," *IEEE Trans. Software Eng.*, vol. SE-9, no. 6, pp. 639-648, Nov. 1983.

[2] J. W. Bailey and V. R. Basili, "A meta-model for software development resource expenditures," in *Proc. 5th Int. Conf. Software Eng.*, 1981, pp. 107-116.

[3] R. D. Banker, "Estimating most productive scale size using data envelopment analysis," *European J. Oper. Res.*, vol. 17, no. 1, pp. 35-44, July 1984.

[4] R. D. Banker, A. Charnes, and W. W. Cooper, "Some models for estimating technical and scale inefficiencies in DEA," *Management Sci.*, vol. 30, no. 9, pp. 1078-1092, Sept. 1984.

[5] R. D. Banker, R. F. Conrad, and R. P. Strauss, "A comparative application of DEA and translog methods: An illustrative study of hospital production," *Management Sci.*, vol. 32, no. 1, pp. 30-44, Jan. 1986.

[6] R. D. Banker and R. C. Morey, "Efficiency analysis for exogenously fixed inputs and outputs," *Oper. Res.*, vol. 34, no. 4, pp. 513-521, July-Aug. 1986.

[7] R. D. Banker and A. Maindiratta, "Piecewise loglinear estimation of efficiency production surfaces," *Management Sci.*, vol. 32, no. 1, pp. 126-135, Jan. 1986.

[8] —, "Nonparametric analysis of technical and allocative efficiencies in production," *Econometrica*, vol. 56, no. 6, pp. 1315-1332, Nov. 1988.

[9] R. D. Banker, "Maximum likelihood, consistency and data envelopment analysis," Carnegie Mellon Univ., Working Paper, 1987.

[10] —, "Stochastic data envelopment analysis," *Management Sci.*, 1989, to be published.

[11] W. A. Barnett and Y. W. Lee, "The global properties of the minflex

---

[15] Helpful comments from C. Beath, R. Davis, and participants at a research seminar at the University of Minnesota on an earlier version of this paper are gratefully acknowledged.

Laurent, generalized Leontief and translog flexible functional forms," *Econometrica*, pp. 1421-1437, 1985.

[12] C. A. Behrens, "Measuring the productivity of computer systems development activities with function points," *IEEE Trans. Software Eng.*, vol. SE-9, no. 6, pp. 648-652, Nov. 1983.

[13] L. A. Belady and M. M. Lehman, "The characteristics of large systems," in *Research Directions in Software Technology*, P. Wegner, Ed. Cambridge, MA: MIT Press, 1979, pp. 106-138.

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

[15] F. P. Brooks, *The Mythical Man-Month*. Reading, MA: Addison-Wesley, 1975.

[16] D. W. Caves and L. R. Christensen, "Global properties of flexible functional forms," *Amer. Econ. Rev.*, pp. 422-432, 1980.

[17] A. Charnes, W. W. Cooper, and E. Rhodes, "Evaluating program and managerial efficiency: An application of data envelopment analysis to program follow through," *Management Sci.*, vol. 27, no. 6, pp. 668-697, June 1981.

[18] L. R. Christensen, D. W. Jorgenson, and L. J. Lau, "Transcendental logarithmic production frontiers," *Rev. Econ. Stat.*, vol. 55, pp. 28-45, 1973.

[19] S. Conte, H. Dunsmore, and V. Shen, *Software Engineering Metrics and Models*. Reading, MA: Benjamin/Cummings, 1986.

[20] T. DeMarco, "Yourdon project survey: Final report," Yourdon, Inc., Tech. Rep., Sept. 1981.

[21] W. Hildenbrand, "Short-run production functions based on microdata," *Econometrica*, vol. 49, no. 5, pp. 1095-1125, Sept. 1981.

[22] D. R. Jeffery and M. J. Lawrence, "An inter-organisational comparison of programming productivity," in *Proc. 4th Int. Conf. Software Engineering*, 1979, pp. 369-377.

[23] C. Jones, *Programming Productivity*. New York: McGraw-Hill, 1986.

[24] G. G. Judge, W. E. Griffiths, R. C. Hill, and T. C. Lee, *The Theory and Practice of Econometrics*. New York: Wiley, 1980.

[25] C. F. Kemerer, "An empirical validation of software cost estimation models," *Commun. ACM*, vol. 30, no. 5, pp. 416-429, May 1987.

[26] H. Varian, "The nonparametric approach to production analysis," *Econometrica*, vol. 52, no. 3, pp. 579-597, 1984.

[27] I. Vessey, "On program development effort and productivity," *Inform. Management*, vol. 10, pp. 255-266, 1986.

[28] C. E. Walston and C. P. Felix, "A method of programming measurement and estimation," *IBM Syst. J.*, vol. 16, no. 1, pp. 54-73, 1977.

[29] C. G. Wingfield, "USACSC experience with SLIM," U.S. Army Inst. for Research in Management Information and Computer Science, Tech. Rep. 360-5, 1982.

Rajiv D. Banker received the B.Sc. degree (with highest honors) in mathematics and economics from the University of Bombay and the doctorate from the Graduate School of Business, Harvard University, Cambridge, MA.

He is the Arthur Andersen Professor of Accounting and Information Systems at the University of Minnesota, Minneapolis. He has authored over 70 papers, including publications in *Management Science*, *Econometrica*, *Operations Research*, and *European Journal of Operational Research*. For the past 10 years he has worked on several research projects involving the design of productivity measurement and management control systems. His current research interests also include the evaluation of economic and strategic impacts of information technologies and software engineering tools.

Dr. Banker is a Co-Editor of the *Journal of Productivity Analysis*, and serves on the editorial boards of five other journals.

Chris F. Kemerer received the B.S. degree in decision sciences and economics from the Wharton School of the University of Pennsylvania, Philadelphia, and the doctorate from the Graduate School of Industrial Administration at Carnegie-Mellon University, Pittsburgh, PA.

He is the Douglas Drane Career Development Assistant Professor of Information Technology and Management at the Massachusetts Institute of Technology, Cambridge, MA. His work has appeared in the *Communications of the ACM* and other journals. His current research interests include software engineering measurement and modeling.

Dr. Kemerer is a member of the IEEE Computer Society, the Association for Computing Machinery, and TIMS.