THE STRUCTURE OF THE SOFTWARE COST FUNCTION[1]

Karen T. Lum, Jairus Hihn

California Institute of Technology/Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109

I. INTRODUCTION

Software costs are a growing concern for many firms. As more and more software is being developed – in-house or contracted out – the proportion of software costs to other costs is growing. Software is a peculiar product with many different characteristics than traditional goods discussed in various economic analyses. An important characteristic of software is that it is primarily labor intensive. Also, because software is an intangible product it is difficult to determine how to measure output. Traditionally, in industry, the output of software is measured by size of the software product – either in quantity of source lines of code or function points. Due to the nature of the Jet Propulsion Laboratory's environment, function point data is not readily available.

Software at NASA and the Jet Propulsion Laboratory can be divided into two main categories, ground systems software and flight systems software. The data being analyzed is mainly ground software. Ground software tends to be evolutionary with updates on older systems, while flight software needs to be more reliable and tends to be highly embedded and more real time than ground software, implying that more experienced programmers are needed. However, these characteristic differences only affect the environmental parameter values. The theoretical structure of the cost model should be applicable for any software with only different slopes and intercepts for each type of software.

From an economic viewpoint, a proper cost function should have the following form:

$$C(w, r, S) = w * L(w, r, S) + r * K(w, r, S) \qquad \text{(Eq. 1)}$$

where C is cost, w and r are prices of inputs L and K respectively, and S is the output quantity. The cost and inputs are functions of the input prices and output. By the principle of duality, a proper cost function can be derived from a production function in the form: $S = f(L, K)$.

[1] This work was performed for the Jet Propulsion Laboratory, California Institute of Technology, sponsored by the National Aeronautics and Space Administration.

where S is the output quantity and L and K are the inputs to production. Theory suggests that the wage rate should be in the cost function. However, many cost estimating relationships (CERs) of commercial cost models do not follow the functional form of Equation 1, suggesting that there may be a fundamental flaw in such models.

There is much literature about software costs, but most use an improper functional form for cost. There has been work to determine whether there are scale economies in software development by Banker and Kemerer (1989). Thibodeau and Dodson (pp.70-78) look at the interrelationships of the software life cycle phases in estimating software costs but from a production theory perspective. Banker, Datar, and Kemerer (1991) have developed an estimable production frontier model of software maintenance. Putnam (pp.167-176, pp.345-61) has done work in analyzing the software equation. But all these past works have focused on the production-side of software development. Those works that have focused on costs use a rearranged production function.

Therefore it is the purpose of this paper to analyze the structure of the CERs that commercial cost models use and to test whether they are viable cost functions. In addition, it is the purpose of this paper to derive the structure of a software cost function from a theoretical production function and to test the significance of such a function against NASA's ground software projects. This work is of greater importance, as software costs are absorbing larger portions of organizations' budgets. It is imperative that the correct structure of the software cost function is known before costs can be estimated.

II. Theoretical Model

Although there are a variety of cost models available commercially, some well known ones are Boehm's Constructive Cost Model (COCOMO), Putnam's Software Life Cycle Management (SLIM), Price-S, and SEER (For further discussion on these models, refer to Ferens and Daly). Many commercial software cost models' algorithms do not take the form of equation 1.

For example, COCOMO II's (Boehm et al.) cost function takes the form:

$$L = AS^{B}E \qquad \text{(Eq. 2)}$$

where L is the level of effort, A is a constant, B is some scaling factor, and E is a group of weighted environmental factors (cost drivers) whose products are used to adjust effort. This equation says that the input L is a function of output and some environmental factors. Equation 2 is merely a rearrangement of some production function. When E is multiplied by the average wage rate, the cost of a software project is obtained in dollars:

$$w*L = C = w*AS^{B}E. \qquad \text{(Eq. 3)}$$

Although the functional form of the COCOMO cost function has the wage rate, it is simply multiplied against labor. The relationship between the wage rate and labor is not being reflected in commercial cost models. A proper econometric cost function should not contain inputs; rather it should contain the prices of inputs. Production functions should contain inputs but not prices of inputs.

The principles of duality can be used to derive the production function implied by the cost function in Equation 2. Based on the structure of the commercial cost model, the production function should be structured as follows:

$$S = (L/AE)^{1/B}. \qquad \text{(Eq. 4)}$$

Equation 4 suggests that commercial cost functions should have the same exponent on all independent variables. The analysis on the implied production function by commercial cost models will be referred to as Model 1.

A cost function can be derived from the production function assuming a Cobb-Douglas format:

$$S = \bar{A}L^{a}K^{b}\bar{E} \qquad \text{(Eq. 5)}$$

where S is the size of software in source lines of code (SLOC), $\bar{A}$ is a constant, L is labor in effort-months, K is capital, and $\bar{E}$ is a set of environmental factors. a and b are elasticity measures. Intuitively, the lines of code should be the output of production. Also, since software is a labor-intensive product, the main input is intuitively labor, which is affected by some environmental factors. Environmental factors may include the capability and knowledge of the development team, required reliability of the software, where the software is being developed, etc. (See Appendix). The constant A is a technology coefficient related to fixed cost.

Assuming that software costs are minimized during production, minimizing

$$C = w*L + r*K \qquad \text{(Eq. 6)}$$

where C is cost, w is wage rate, L is labor in effort-months, r is cost of capital, and K is capital and constraining it against the production function (Eq. 5) gives the following empirical cost equation:

$$C = (S/\bar{A}\bar{E})^{1/(a+b)}w^{a/(a+b)}r^{b/(a+b)}\,[(a/b)^{b} + (a/b)^{-a}]^{1/(a+b)}. \qquad \text{(Eq. 7)}$$

Assuming a Cobb-Douglas production function leads us to a Cobb-Douglas cost function.

There is no data on price of capital. r. However, it can be assumed that r =1 because as technology increases, the cost of technology remains relatively constant. With Cobb-Douglas functions, units are usually

measured so that $\bar{A} = 1$ (Varian, p.65). We can simplify Equation 7 by calling the term $[(a/b)^b + (a/b)^{-a}]^{1/(a+b)}$ as $\bar{A}$':

$$C = \bar{A}' w^{a/(a+b)} (S/\bar{E})^{1/(a+b)}. \qquad \text{(Eq. 8)}$$

The structure of Equation 8 suggests that labor needs to be a function of the wage rate, which many commercial cost models fail to consider. Since the wage rate did not cancel out or could not be isolated while it was being derived from the production function, it has some significance in estimating labor effort. This model allows for both increasing and decreasing returns to scale by not restricting a + b to equal 1, as Banker and Kemerer (1989) have stated that the existence of local scale economies or diseconomies depends on the size of software development projects. The analysis on the derived cost function will be called Model 2.

III. DATA

This study uses data from NASA's historical database 1986-1990 and from the Jet Propulsion Laboratory 1988-1990. There are 60 ground software projects with data on labor, size of code, and environmental drivers. This data can be used to test the significance of a suggested production function. Cost and wage data was not available for all 60 observations. Therefore, the sample has been reduced to 43 data points for estimating the cost function. Table 1 and Table 2 summarize the data.

Data for environmental factors was collected using COCOMO 81's fifteen cost drivers (See Appendix). These fifteen environmental factors were determined by Boehm as significant and independent (Boehm, 1981). Interviewees from the 60 ground software projects were asked to rate the fifteen environmental drivers on a scale of very low to high. Each environmental factor can affect costs by driving it up or down from the nominal, where nominal ratings have the value of 1. These qualitative ratings have assigned values, which are multiplied against each other to determine how labor should be adjusted:

$$E = \prod_i EM_i = EM_1 * EM_2 * EM_3 * \cdots * EM_{15} \qquad \text{(Eq. 9)}$$

where i = 1 through 15.

Although the inputs into software include labor and capital as well as some environmental factors, basic capital costs were grouped with the labor factors. Part of the cost of capital is embedded in the wage rate in the way NASA performs its bookkeeping. Labor or effort is measured in number of person months – the amount of time a

person spends working on the software development project for one month excluding time for holidays, vacations, and weekends. The price of labor or the wage rate includes burden for basic capital that a programmer would require as part of his/her task. Therefore the labor term includes basic capital (desktop computer, telephone and electronic mail services, etc.) and will be referred to as the Labor Set, L, with the price of the Labor Set, w. There exists other capital such as special development tools that would be priced separately, but since data for this type of capital cannot easily be obtained, it will be assumed that the costs of these tools are insignificant to the overall cost of software production. To better understand the impact of tools on software production, future work in this area will be considered. Data on other costs of non-basic capital was not readily available, but it can be assumed that as time passes, the cost of capital is constant because as technology becomes faster in terms of processing speed and memory, the costs of this capital for producing software drops. Therefore the other part of capital will be assumed away because it can be understood that the price of capital, r, equals 1. A disturbance term ε should capture any error effects from the assumption that $r = 1$

	L (person-months)	*E*	*S (KSLOC)*
Mean	406.41	0.94	74.59
Standard Error	84.81	0.04	12.54
Median	118.8	0.88	30.5
Mode	60	0.88	100
Standard Deviation	656.97	0.30	97.1721
Sample Variance	431603.9	0.09	9442.415
Range	3231.6	1.49	420.8
Minimum	8.4	0.38	2.2
Maximum	3240	1.87	423
Sum	24384.8	56.59	4475.3

TABLE 1. SUMMARY STATISTICS FOR DATASET 1

	S (KSLOC)	*COST ($)*	*w*	*E*	*L*
Mean	53.21	1217461.43	4551.07	0.88	237.39
Standard Error	9.79	251391.93	84.86	0.04	50.43
Median	24.6	420906	4483	0.88	82
Mode	100	241332	4483	0.88	60
Standard Deviation	64.22	1648487.112	556.49	0.25	330.67
Sample Variance	4124.22	2717510000000	309678.34	0.06	109341.46
Range	279.9	7175291.74	2072.20	0.94	1359.60
Minimum	2.2	37786.87	3411.31	0.38	8.4
Maximum	282.1	7213078.608	5483.51	1.32	1368
Sum	2287.9	52350841.31	195695.9	37.96	10207.6
Count	43	43	43	43	43

TABLE 2. SUMMARY STATISTICS FOR DATASET 2

The software project cost data does not include the cost of large capital expenditures for tools and other materials. The size of software is measured in thousands of source lines of code (KSLOC).[2] The wage rate is the average monthly wage rate of a senior engineer in real year dollars. Historical data on wage rates was not available. The wage rates were derived by taking the average monthly wage rate of a senior engineer in fiscal year 2002 and deflating it by the NASA accumulated inflation index for the appropriate year to get monthly wage in real year dollars. The wage rates were then averaged over the life of specific projects.

IV. EMPIRICAL STRATEGY

To analyze Model 1, Equation 4 can be linearized by taking the logarithms of both sides:

$$\ln S = -(1/B)(\ln A) + (1/B)(\ln L - \ln E) + \varepsilon. \qquad \text{(Eq. 10)}$$

The term $-(1/B)(\ln A)$ will generate a regression result β_0 and the coefficient on $(\ln L - \ln E)$ will generate a regression estimate β_1:

$$\ln \hat{S} = \beta_0 + \beta_1(\ln L - \ln E). \qquad \text{(Eq. 11)}$$

If $\beta_1 = 0$ then the structure of commercial production functions are unsound.

To estimate whether wage rate is significant to the cost function for Model 2, the logarithms of both sides of equation 8 is taken, and then a linear model is estimated using regression techniques

[2] Although the actual lines of code can be used instead of thousands of lines of code, it is the industry standard to measure lines of code in thousands. It should be noted that this is only a change in scale and should not affect the significance of the analysis. The logarithm of KSLOC will generate a regression result with a negative intercept, while the logarithm of SLOC will produce a positive intercept.

$$\ln C = \ln \bar{A}' + [a/(a+b)]\ln w + [1/(a+b)](\ln S - \ln \bar{E}) + \varepsilon. \qquad \text{(Eq. 12)}$$

A regression on the dependent variable ln C will generate an estimated function:

$$\ln \hat{C} = \delta_0 + \delta_1 \ln w + \delta_2(\ln S - \ln \bar{E}) \qquad \text{(Eq. 13)}$$

where δ_0 is the estimated intercept ln $\bar{A}'$, δ_1 is the estimated coefficient [a/(a+b)], and δ_2 the estimated coefficient [1/(a+b)].

If the wage rate is insignificant to the cost function δ_1 should be 0, indicating that the wage rate does not need to be in the cost equation, Equation 13 can be rewritten

$$\ln \hat{C} = \delta_0 + \delta_2(\ln S - \ln \bar{E}). \qquad \text{(Eq. 14)}$$

Taking the inverse log of Equation 14, gives

$$C = \bar{A}'(S/\bar{E})^{\delta_2}. \qquad \text{(Eq. 15)}$$

Notice that Equation 15 does not have the wage rate in it. If δ_1 equals 0, then it means that the commercial cost models can develop cost equations by simply rearranging the production function and multiplying it by the wage rate to obtain software costs. But if δ_1 is not equal to 0 then the wage rate in Equation 8 is significant and cannot be taken out of consideration when developing cost models.

V. Statistical Analysis

Ordinary least squares regressions were run on the datasets. Two regressions were run on Model 1 to test the validity of the implied software production function from the functional form of commercial cost models. One regression was performed on dataset 1 and another on dataset 2. The results are presented in Table 3.

Table 3. OLS Estimates for Production Function Parameters (Model 1) Dependent variable: ln S

	Dataset 1 (n = 60) F-value = 1134.66 Adj. R^2 = .9505		Dataset 2 (n = 43) F-value = 664.3 Adj. R^2 = .9405	
Independent variable	Coefficient	Standard error	Coefficient	Standard error
constant	-1.032169058*	0.140668	-1.11388*	0.177501
ln L – ln E	0.883726955*	0.026235	0.907588*	0.035213

* significant at the 1% level

The F-values for both Datasets in testing Model 1 are significantly high suggesting that the model is understood. The adjusted R^2 values are close to 1 for both datasets. The coefficients of Model 1 for both data sets are

statistically significant at the 1% level. The production function implied by many commercial cost models is understood. However, this does not mean that it is the best model. Another model may be even more useful in terms of providing more reliable estimates and predictions (McClave, et al., p. 558). NASA's estimated production function implied from commercial cost functions is

$$S = 2.8(L/E)^{0.884} \qquad \text{(Eq. 16)}$$

from dataset 1, and

$$S = 3.05(L/E)^{0.908} \qquad \text{(Eq. 17)}$$

from dataset 2. The exponents on equations 16 and 17 represent the 1/B in equation 4.

In addition, Equation 3's functional form was also tested for significance. The regression results are in Table 4.

TABLE 4. OLS ESTIMATES FOR COMMERCIAL COST FUNCTION DEPENDENT VARIABLE: ln C

	Dataset 2 (n = 43) F-value = 270.87 Adj. R^2 = .9507	
Independent variable	Coefficient	Independent variable
constant	3.079367	2.895252
ln E	0.546567*	0.136755
ln S	1.029535*	0.036648
ln w	0.815328**	0.344801

*significant at the 1% level
**significant at the 5% level

Although the commercial cost model CER appears to be understood, its F-value is lower than for the F-values on the other functional forms tested. If the coefficient of ln w were 0, it would suggest that wage does not affect labor and therefore it could be multiplied in to get cost as in equation 3. The exponent on ln w is significantly greater than 0, suggesting that the wage rate is associated to cost and should be a variable in equation 2

$$L = AS^{B1}w^{B2}E. \qquad \text{(Eq. 18)}$$

An estimated exponent value for ln S, B_1, 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

$$\rho = (S/L) * (dL/dS) = B_1 \qquad \text{(Eq. 19)}$$

That is, marginal productivity (dS/dL) is greater than average productivity (S/L) if B_1 is less than 1 (Banker and Kemerer, 1989). This cost function suggested by commercial cost models implies that there are diseconomies of scale in software development at NASA.

One regression was run on Model 2 using data from dataset 2 to test the structure of the Cobb-Douglas software cost function. The results are presented in Table 5.

TABLE 5. OLS ESTIMATES FOR COST FUNCTION PARAMETERS (MODEL 2) DEPENDENT VARIABLE: ln C

	Dataset 2 (n = 43) F-value = 5967.84 Adj. R^2 = .9965	
Independent variable	Coefficient	Standard error
constant	-20.20412871*	0.636627
ln w	12.9433461*	0.271545
ln S – ln E	0.019341082	0.020801

* significant at the 1% level

The extremely high F-value of 5967.84 suggests that the model is statistically useful. The adjusted R^2 is very close to 1, indicating that almost all of the sample variation in cost is explained. t-tests on the independent variables suggest that there is sufficient evidence that the wage rate is related to cost. On the other hand, the coefficient on the ln (S/E) term is not statistically significant suggesting that S and E should not share the same exponent. This regression suggests that NASA's software cost equation should be

$$C = 0.00000000169w^{12.94}(S/\bar{E})^{0.019} \qquad \text{(Eq. 20)}$$

intercepting near the origin.

If the coefficient of ln w is equal to 0, then the function suggests that Equation 15 is a proper cost function. However, the test of null that wage is insignificant to cost

$$H_0: \delta_1 = 0$$

is rejected at the 1% significance level. We accept the alternative hypothesis that wage is significantly related to cost.

VI. CONCLUSION

Regression results suggest that the cost model derived from the Cobb-Douglas production function in Model 2 is understood. The F-test is significant. t-tests on the variable coefficients are also significant suggesting that the Cobb-Douglas structure of the cost model is appropriate.

The wage rate is a necessary variable in a proper software cost function. This can be a result of software being a highly labor-intensive product. Also, labor is a function of the wage rate, suggesting that the level of productivity or lines of code produced per person-month (S/L) is related to how much they get paid.

There is insufficient evidence to disprove the functional form of commercial cost models. Yet, theory implies that the wage rate is endogenous to labor, and commercial cost functions do not capture this feature. A problem lies in the structural form of commercial cost models. They do not take into account that labor can be affected by wage. Therefore Equation 2 should look more like the following:

$$L = AS^{B1}w^{B2}E \qquad \text{(Eq. 21)}$$

and a cost function should have the following structure

$$C = w*L = AS^{B1}w^{B2+1}E \qquad \text{(Eq. 22).}$$

The production function tested in Model 1 suggests that commercial cost functions are adequate for predicting and estimating costs, although they are not as strongly significant as other cost function structures that inherently include the wage rate.

Although the two different models produced mixed results, this study implies that the functional form of the software cost equation *can* be different. A concern in this analysis is the lack of data on capital prices and inputs. Although it was the assumption of this work that technology prices remain relatively constant over time and that r = 1, this may not be the case. While only a Cobb-Douglas production and cost function were analyzed in this study, it is worthwhile to consider other higher order functional structures in future work.

APPENDIX

Software Development Cost Drivers

		Ratings					
Drivers	Symbol	Very Low	Low	Nominal	High	Very High	Extra High
RELY Required Software Reliability	EM_1	0.75	0.88	1.00	1.15	1.40	
DATA Database Development Size	EM_2		0.94	1.00	1.08	1.16	
CPLX Product Complexity	EM_3	0.70	0.85	1.00	1.15	1.30	1.65
TIME Execution Time Constraint	EM_4			1.00	1.11	1.30	1.66
STOR Main Storage Constraint	EM_5			1.00	1.06	1.21	1.56
VIRT Virtual Machine Volatility	EM_6		0.87	1.00	1.15	1.30	
TURN Computer turnaround time	EM_7		0.87	1.00	1.07	1.15	
ACAP Analyst Capability	EM_8	1.46	1.19	1.00	0.86	0.71	
AEXP Applications Experience	EM_9	1.29	1.13	1.00	0.91	0.82	
PCAP Programmer Capability	EM_{10}	1.42	1.17	1.00	0.86	0.70	
VEXP Virtual Machine Experience	EM_{11}	1.21	1.10	1.00	0.90		
LEXP Programming Language Experience	EM_{12}	1.14	1.07	1.00	0.95		
MODP Use of modern programming practices	EM_{13}	1.24	1.10	1.00	0.91	0.82	
TOOL Use of Software Tools	EM_{14}	1.24	1.10	1.00	0.91	0.83	
SCED Required Development Schedule	EM_{15}	1.23	1.08	1.00	1.04	1.10	

Boehm, B. (1981). *Software Engineering Economics.* New Jersey: Prentice-Hall, Inc.

REFERENCES

R. Banker and C. Kemerer, "Scale Economies in New Software Development," *IEEE Transactions on Software Engineering,* vol. 15, no. 10, pp.1199-1205, Oct. 1989.

R. Banker, S. Datar, and C. Kemerer, "A Model to Evaluate Variables Impacting the Productivity of Software Maintenance Projects," *Management Science*, vol. 37, no. 1, pp. 1-18, Jan. 1991.

B. Boehm, *Software Engineering Economics*, Englewood Cliffs, New Jersey, Prentice-Hall, Inc: 1981.

B. Boehm, et al., *Software Cost Estimation with COCOMO II*, Upper Saddle River, New Jersey, Prentice Hall PTR: 2000.

D. Ferens and B. Daly, "A Quantitative Comparison of Popular Software Scheduling Models," *1991 ISPA Conference Proceedings*, vol. X, pp. SW43-SW59.

J. McClave, P. Benson, and T. Sincich, *Statistics for Business and Economics* (8th ed.), Upper Saddle River, NJ, Prentice Hall: 2001.

L. Putnam, "The Real Economics of Software," *The Economics of Information Processing*, vol. 2, pp.167-176.

L. Putnam, "A General Empirical Solution to the Macro Software Sizing and Estimating Problem," IEEE *Transactions on Software Engineering*, 4(4), 345-61, 1978.

R. Thibodeau and E. Dodson, "The Implications of Life Cycle Phase Interrelationships for Software Cost Estimating," *General Research Corporation, Economic Resources and Planning Operations*, pp. 70-76.

H. Varian, *Microeconomic Analysis* (3rd ed.), New York, W. W. Norton & Company: 1992.