Explaining the Cost of European Space and Military Projects

Lionel C. Briand, Khaled El Emam, Isabella Wieczorek
Fraunhofer Institute for Experimental Software Engineering
Sauerwiesen 6
D-67661 Kaiserslautern
Germany
{briand, elemam, wieczo}@iese.fhg.de

International Software Engineering Research Network Technical Report ISERN-98-19

Explaining the Cost of European Space and Military Projects

Lionel C. Briand, Khaled El Emam, Isabella Wieczorek
Fraunhofer Institute for Experimental Software Engineering (IESE)
Sauerwiesen 6
D-67661 Kaiserslautern
Germany
+49 6301 707 251
{briand,elemam,wieczo}@iese.fhg.de

ABSTRACT
There has been much controversy in the literature on several issues underlying the construction of parametric software development cost models. For example, it has been argued whether (dis)economies of scale exist in software production, what functional form should be assumed between effort and product size, whether COCOMO factors were useful, and whether the COCOMO factors are independent. Answers to such questions should help software organizations define suitable data collection programs and well-specified cost models. The only way to address these issues and obtain a generalizable conclusion is to investigate them on a large number of consistent data sets. In this paper we use a data set collected by the European Space Agency to perform such an investigation. To ensure a certain degree of consistency in our data, we focus our analysis on a set of space and military projects that represent an important application domain and the largest subset in the database. These projects have been performed, however, by a variety of organizations. First, our results indicate that two functional forms are plausible between effort and product size: linear and log-linear. This also means that different project subpopulations are likely to follow different functional forms. Second, besides product size, the strongest factor influencing cost appears to be team size. Larger teams result in substantially lower productivity, which is interesting considering this attribute is rarely collected in software engineering cost data bases. Third, although some COCOMO factors appear to be useful and significant covariates, they play a minor role in explaining project effort. Overall, the most plausible model appears to be a log-linear model involving KLOC, team size, and a principal component influenced by three COCOMO factors: reliability requirements (RELY), storage constraints (STOR), and execution time constraints (TIME). High values for these factors are likely to be associated with embedded systems, which usually share these characteristics.

Keywords
Software Cost Estimation, Model Specification, Economies of Scale

1 INTRODUCTION

There has been keen research interest in developing a general theory of software development resource expenditures. Such a theory would take the form of the relationship(s) between product size, development effort, and productivity factors. This is evidenced by the recent large scale effort to develop the COCOMO II model [6], as well as the large number of earlier studies [3][4][5][12][13][16][17][19]. It is believed that such a theory would alleviate the current software production inefficiencies and cost overruns through a better understanding of factors affecting cost, of how to manage projects to maximize productivity, and by providing improved cost estimation capabilities.

Important ingredients of such a theory are (1) an elaboration of whether economies or diseconomies of scale exist in software production, and (2) the exact nature of the functional form of the effort/size relationship (e.g., linear, quadratic, exponential, log-linear, or translog).

There have been a series of studies that investigate whether (dis)economies of scale exist in software production [1][2][17], and the functional form of the relationship between effort and size [12]. However, taken as a whole, these studies provide an inconsistent picture, due in part to the use of different data sets from different application domains, but also to different analysis techniques. This makes it difficult to make general statements on whether there indeed are any (dis)economies of scale, and what the functional form of the effort-size relationship should be.

In this paper we investigate the existence of (dis)economies of scale, and the functional form of the effort-size relationship. We do so within a single application domain (space and military projects) with a large multi-organizational database, using the same analysis techniques as previous researchers to promote comparability of results.

The database projects come from a variety of European countries and have been contracted by the European Space Agency (ESA). Since this database covers a large number of organizations and represents a significant portion of the market in the application domains considered, we expect that our findings are of reasonable external validity, i.e., can be generalized to the European space and military application domain at large.

Briefly, our results show that the relationship between effort and size appears to be log-linear (i.e., of the form $a \times KLOC^b$). Furthermore, we did find economies of scale in this data set. It was also found that, like in other studies [7][14][22], higher team size results in lower productivity due in part to larger communication overhead. This is consistent with the underlying assumptions of the COCOMO and Putnam models. Finally, some of the productivity factors (a subset of the COCOMO factors) collected here appear to be significant covariates in the effort models. However, their impact is relatively weak and they do not substantially improve the models' goodness of fit. This may be explained by the fact we are working in a well-defined application domain where such factors do not strongly discriminate between projects.

The practical implications of these results for the European space and military software industry are threefold. First, it suggests that if projects are divided up into multiple independent increments, one may assess the productivity losses associated with such a decision and weigh it against the gains expected from incremental development. Secondly, the impact of team size observed on this data set is very substantial, suggesting that such a measure should be systematically and carefully collected in software engineering cost databases. Finally, the weak effect of the COCOMO factors that were considered suggests that organizations should not directly use the generic productivity factors, but ought to consider identifying their own productivity factors, or at least customizing the generic productivity factors to their local environment.

The structure of the paper is as follows. After the Introduction, Section 2 discusses the concept of economies of scale and different functional forms capturing the relationship between effort and size. It is followed by a discussion of interrelationships between COCOMO factors. Section 3 describes the database we used for our analysis, the measures we used to compare the different functional forms, how we accounted for cost factors in our models, and how we dealt with outliers in the data. Section 4 provides the analysis results in terms of univariate and multivariate regression models, as well as their comparison. Finally, Section 5concludes the paper and gives some information about our future research directions.

2 BACKGROUND

Economies of Scale and Functional Form

The concept of economies of scale states that average productivity increases as the system size increases. This has been attributed, for example, to software development tools whereby the initial tool institutionalization investment may preclude their use on small projects [5]. Furthermore, there may be fixed overhead costs, such as project management, that do not increase directly with system size, hence affording the larger projects economies of scale. On the other hand, it has been noted that some overhead activities, such as documentation, grow at a faster rate than project size [14], contributing to diseconomies of scale. Furthermore, within a single organization, it is plausible that as systems grow larger, then larger teams will be employed. Larger teams introduce inefficiencies due to an increase in communication paths [7], the potential for personality conflicts [5] and more complex system interfaces [8].

The four functional forms that have been investigated in the literature for modeling the relationship between system size and effort are summarized in Table 1. It is clear that the linear model does not exhibit any (dis)economies of scale. A popular functional form for this relationship has been the log-linear model, specified by the well-known COCOMO model [5].

Kitchenham [17] looks at how significantly different from 1 is the exponent parameter in the log-linear model (the *b* parameter in Table 1). She concludes that, over 12 datasets, the relationship between effort and size is rather linear since most coefficients are not significantly different from 1. Banker and Kemerer [1] use the econometric concept of elasticity [15] in order to determine whether there is an ideal project size (the "most productive scale size" or MPSS) where productivity is optimal. Over 9 datasets, they show that, although MPSS shows large variation, there was evidence of both economies and diseconomies of scale. They conclude that traditional models such as the log-linear relationship are therefore too limited to take into account the effort/size relationship. More recently, Hu [12] revisited some of the data sets already investigated and concluded that, over 9 data sets, the quadratic model seemed to be the most plausible relationship between effort and size in comparison with the other three. The comparison procedure he used, referred to as the P-test, was designed to test the specifications of econometric models since, he argues, non-nested models cannot be compared by just looking at the adjusted R^2.

Hu's [12] results are promising since they are based on an objective and statistically valid approach for *comparing* the four different functional forms. However, the conclusions that are drawn have a number of important weaknesses. First, no outlier analysis was performed on the data sets that were used. It is well known that, at least for one of the data sets coming from the work of Kemerer [13] there is one extreme outlier that has a substantial influence on the results of regression analysis (see [18]). Second, many of

the data sets are quite old, some dating from the late seventies and early eighties. It is not clear that the same phenomena would be observed in modern software production. Third, an analysis that was used to support the conclusions involves the pooling of eight different data sets together. This is highly questionable since there would be inconsistencies in the manner in which both effort and size (in Lines of Code) are measured across these data sets. Finally, there are inconsistent arguments presented in justifying the conclusions. For instance, the results of an analysis of the Kemerer data set is used to justify the strong conclusion that the linear model should be rejected as a plausible functional form. Note that the Kemerer data set has 15 projects. However, on the same page, the results of the analysis of the Wingfield data set that show that the linear model is a more plausible functional from than the quadratic model are discounted, because "being one of the smallest data set (15 observations), the significance of these results should be discounted".

Clearly then, it is imperative to continue studying the functional form of the relationship between effort and size.

Model Specification	**Model Name**
$Effort = a + (b \times Size)$	Linear Model
$Effort = a + (b \times Size) + (c \times Size^2)$	Quadratic Model
$Effort = e^a \times Size^b$	Log-linear Model
$Effort = e^a \times Size^b \times Size^{c \times \ln Size}$	Translog Model

Table 1: Different functional forms for modeling the relationship between effort and size.

Interrelationships between COCOMO Factors

COCOMO-based cost estimation models assume that the factors (cost drivers) are independent of one another. However, several studies demonstrated that the cost factors are often interrelated. Kitchenham [16] shows that there is a relationship between two of the COCOMO factors based on the COCOMO data set itself. Similarly, through principal component analysis, Kitchenham [17] found that, out of 21 cost factors, seven principal components accounted for 75% of the effort variability in a data set of 28 projects. This supports the results of Subramanian et. al. [23]. Through factor analysis of the COCOMO data set, they reduced the 15 COCOMO factors to four factors, accounting for 73% of variation in the variable space. The identified concepts are expressed as constraints: application constraint, virtual machine and language constraint, completion within schedule constraint, and programming capability constraint. Maxwell et. al. [19] analyzed the ESA database, including projects from the space, military, and industrial environments. They report that the seven collected COCOMO factors could be grouped into four factors explaining 90% of the variance in the data. The first factor included TIME, STOR, and RELY, the second factor was MODP and TOOL, the third factor consisted of LEXP, and the fourth was VIRT (see Table 2 for variable description).

The studies presented above suggest that COCOMO factors do not capture independent concepts. In the remainder of this paper, we will therefore perform principal components analysis in order to identify the underlying concepts captured by the COCOMO factors in our data set and use its results to help interpret the results of our analysis.

3 RESEARCH METHOD

In this section, we describe the data set we have used to perform this research. Then, the method used to compare alternative cost models is presented.

Data Source

The database used in this study is the European Space Agency (ESA) multi-organization software project database. Since 1988, the ESA continuously collects historical project data on cost and productivity from different application domains. The data comes from European organizations, with applications from the aerospace, military, industrial, and business environment. Each data supplier is contacted on a regular basis to determine if projects are nearing completion. Once a project questionnaire is filled out, each data supplier is contacted to ensure the validity and comparability of the responses. Each data supplier regularly receives data analysis reports of the data set.

At the time of our analysis, the database consisted of 158 projects. The breakdown of projects by environment was: 36% space, 32% military, 22% business, and 10% industry projects. The variables that are taken into account in our analysis are listed in Table 2. These are variables that potentially may have an impact on software project cost.

Because the database contains projects that used multiple programming languages, we limited our analysis to the 64 projects from the space and military environment developed with high-level languages, leaving out projects developed (partly or fully) with Assembler, for example. This increases our confidence that we have somewhat comparable size measurement.

Comparison of Models

Functional Forms to be Compared

We fit our data to each of the four models in Table 1 using linear ordinary least squares regression. The bottom two models are appropriately converted into a linear model, in order to allow the use of linear least-squares regression estimates. Certain precautions were taken to address the existence of outliers, which is a common occurrence with cost and productivity data. These are explained below.

Comparing the Models' Goodness of Fit

The two traditional ways of assessing and comparing the

Variable	Description	Scale	Values / Range / Unit
PROJTYPE	Type of SW Project	nominal	Customized Application, Partly Customized Application, Integration Project, Embedded Application, SW Product Development, Other
KLOC	New developed code	ratio	1 KLOC=1000 LOC
EFFORT	Effort for SW project	ratio	Person hours , where 144 person hours=1 person month
TEAM	Maximal team size on one stage of a project	ratio	
VIRT	virtual machine volatility	ordinal	2-5 (low-very high)
RELY	required reliability	ordinal	1-5 (very low-very high)
TIME	execution time constraints	ordinal	3-6 (nominal-extra high)
STOR	main storage constraint	ordinal	3-6 (nominal-extra high)
MODP	use of modern programming practices	ordinal	1-5 (very low-very high)
TOOL	use of software tools	ordinal	1-5 (very low-very high)
LEXP	programming language experience	ordinal	1-4 (very low-high)

Table 2: Variables from the ESA Database

goodness of fit of cost models is to compute their coefficient of determination R^2 (adjusted for the number of variables), their Pred (.25) value, and their mean magnitude of relative error (MMRE) [8]. However, although it may be an important selection criterion for a model, the MMRE is sensitive to the weight that a model grants to smaller projects. Because of the MRE's mathematical formulation smaller projects tend to yield the larger MRE values. For example, the log-linear model, because of its logarithmic transformation, grants more weight to smaller projects than the linear model. The MRE values for the smaller projects become lower through the logarithmic transformation and therefore, the MMRE is usually smaller. But, it is important to note that a smaller MMRE does not indicate in any way that the log-linear model is a more plausible alternative.

In our study we perform two kinds of model comparisons which should be distinguished since they require different comparison techniques. In the simplest case, we compare nested models: one model has a set of terms which is a subset of the other model's terms, e.g., linear and quadratic models. In this case, the adjusted R^2 can simply be used as a means of comparison. A second, more complicated case, is when the two models to be compared are not nested. As described in [12], non-nested models cannot be compared using the R^2 since this one is affected by the use of different variables, showing different spacing within the data. Therefore, as suggested by Davidson and MacKinnon [10] and used by Hu [12], a series of tests can be used to test whether a given model is the most plausible among several alternatives.

Identifying the Most Plausible Models

Davidson and MacKinnon [10] propose a set of tests which are easier to use in different circumstances. Although Hu [12] used the P-test in his study, it is recommended to use the J-test when testing the plausibility of linear (or linearized) models. The J-test is easier, more intuitive, and should yield identical results. When comparing two models, the J-test consists of performing the following regression:

$$y_i = (1-\lambda)\times f_i(X_i,\beta) + \lambda \times \hat{g}_i + \varepsilon_i,$$

$\hat{g}_i = g_i(Z_i,\hat{\gamma})$, where $\hat{\gamma}$ is the estimate of γ,

where

$$H_0 : y_i = f_i(X_i,\beta) + \varepsilon_{0i},$$

y_i is the i[th] observation on the dependent variable, X_i is a vector of observations on independent variables, β is a vector of parameters to be estimated**,** and the error term is assumed to be normally distributed.

$$H_1 : y_i = g_i(Z_i,\gamma) + \varepsilon_{1i},$$

Z_i is a vector of observations on independent variables, γ is a vector of parameters to be estimated**,** and the error term is assumed to be normally distributed. Using the formula above, assuming we wish to test that f_i is the most plausible model, then we test whether λ is equal to zero. If this is the case, then the alternative model is not needed to explain variations in effort. This test can be performed using the usual two tailed t-test based on the λ estimate and its standard error in order to determine whether the λ estimate is significantly different from zero. To test the plausibility of g_i, the two functions just have to be substituted in the formula above and the t-test performed again**.** If the two t-tests, for the two alternative models, tell us that λ is not significantly different from zero, then both models are plausible. If one t-test shows an λ value

significantly different from zero, whereas the t-test for the alternative model does not, then the former model is less plausible than the latter one.

For the quadratic and translog models, we determined whether the regression coefficients of the quadratic terms of the equations are statistically significant. If not, then only the linear and log-linear models, respectively, have to be considered for the J-test. For the log-linear model, the exponential term coefficient is also tested in order to determine whether it is significantly different from 1 (and not 0, as the usual t-test procedure goes). This will confirm whether (dis)economies of scale are plausible, based on the data.

Accounting for COCOMO Factors

It has been shown in the literature that the COCOMO factors are often interrelated. In order to facilitate the interpretation of our results, we perform principal components analysis [11] to determine the actual underlying concepts of the COCOMO factors measured in this data set. Then, the resulting principal components are used instead of the COCOMO factors themselves in order to improve the fit of the cost models. Team size is also used independently as a covariate in the cost model equations. Using stepwise regression, we identify the best multivariate regression models and assess the relative impact of each factor (KLOC, team size, COCOMO principal components) on effort. In the log-linear (and translog) model, these factors have implicitly a multiplicative effect on effort whereas in the linear (and quadratic) model, they have an additive effect. We have also considered interaction terms in the equations in order to take into account interaction effects between KLOC and the other factors. In the log-linear model, significant interaction terms would mean that the extent of the economies of scale is affected by the factors themselves.

Outliers

Outliers (i.e., overinfluential data points in this context) can have a substantial impact on the results of a least squares regression analysis, e.g., R^2, coefficients. For software cost and productivity data, it is common to see many projects with effort towards the low end of the scale and then a few very large projects or projects with extraordinarily high productivities. This may be due to inconsistencies in measurement or to the fact that a few projects belong to a different statistical population than the rest of the data set. Outlying observations can pull the regression plane towards them to optimize the squared error criterion, but this results in models that are not stable. If the outlying observation is removed then dramatically different results would emerge. Furthermore, when assessing the goodness of fit using the mean MRE, the MRE values tend to be inflated because many of the small projects exhibit relatively large MRE's.

Not dealing with outliers adequately can produce misleading results, and has done so in the past in the cost estimation domain (see [17]). It is therefore prudent to consider this particular issue during our analysis. As criteria to identify outliers, we used the widely used Cook's distance [9] and R^2 measures. In a stepwise manner, we removed each observation with the highest Cook's distance measure, until a certain stability was achieved in terms of R^2. This allows us to ensure that the results we obtained were not due to a few observations but were representative of the general trends in our data set.

4 RESULTS

The first subsection focuses on building effort models capturing the relationship between effort and product size, as well on the impact of additional factors such as team size and several COCOMO factors. The second subsection discusses the results in general.

Model Specification	**Parameter Estimates**		**Std Error**	**R^2 / R^2 adj**	**MMRE**	**Pred(.25)**	**Obs**
$Effort = a + (b \times KLOC)$	a (not sign.)	6447.69	3364.5	0.41 / 0.40	1.24	19%	64
	b	323.17	49.17				
$Effort = a + (b \times KLOC) + (c \times KLOC^2)$	a	5515.07	4121.57	0.41 / 0.39	1.19	22%	64
	b	371.98	132.46				
	c (not sign.)	-0.22	0.56				
$\ln(Effort) = a + (b \times \ln(KLOC))$	a	7.02	0.26	0.42 / 0.41	0.69	27%	64
	b	0.73	0.082				
$\ln(Effort) = a + (b \times \ln(KLOC)) + c \times (\ln(KLOC))$	a	7.38	0.54	0.42 / 0.41	0.67	26%	64
	b (not sign.)	0.45	0.37				
	c (not sign.)	0.045	0.057				

Table 3: Types of models: relationship Effort vs. System size

Building Effort Models

Relationship between Effort and Size

The results in Table 3 summarize the four models that were developed between effort and size utilizing the four considered functional forms. All models have an R^2 value that is statistically significant at an alpha level of 0.05. In

general, all the four models explain approximately 40% of the variation in these projects. The quadratic model had a *c* parameter that was not statistically significant. The translog model had *b* and *c* parameters that were both not statistically significant. This indicates that these two functional forms do not provide adequate models explaining effort since the parameters of their quadratic terms are not significantly different from zero.

We are left with the linear model and the log-linear model as plausible alternatives for modeling the relationship between effort and size. For these two models, the relevant parameters are statistically significant. Furthermore, the log-linear model has a *b* parameter that is different from 1 beyond what would be expected by chance (at a two-tailed alpha level of 0.05). This indicates the potential existence of economies of scale, and also that the linear and log-linear models capture two significantly different relationships between effort and size.

It is interesting to compare these findings with previous research. First, unlike the conclusions of Hu [12], we do not find the quadratic model to provide the best specification of the relationship between effort and size. We found the parameters in this model to be essentially zero. Furthermore, our results are different from those of Kitchenham [17] in that the exponent for the log-linear model was found to be different from one, indicating economies of scale. However, our results so far are also similar to Kitchenham's in that we do find the linear model as a plausible alternative.

There have been discussions in the literature about the confusion caused by inconsistent results on the existence of economies/diseconomies of scale [1]. This confusion is because different studies conclude that there are different laws relating size and effort. Our results show that, by looking at only size and effort, indeed different laws are plausible: linear and log-linear.

H_0	H_1	p-value for λ
Linear model	Log-linear model	0.4034
Log-linear model	Linear model	0.44

Table 4: J-test results for model 1 and model 3 from Table 3

One approach for comparing these two different laws is the J-test. The results from the application of the J-test are shown in Table 4. None of the comparisons yield a statistically significant result, confirming that the two laws are both plausible based on our data set.

It is important to also note that the approach proposed by Banker and Kemerer [1] was tried in order to determine if there were a most productive scale size (MPSS). However, since the relationships appeared to be rather exponential or linear, it expectedly did not yield any result. It is possible that such a MPSS would only be visible in a data set coming from one organization.

The question then becomes whether we can improve on these two plausible models to find a better law and also to explain more of the variation in effort. The latter is clearly important because the effort and size models can explain only 40% of the variation in effort, and also it is intuitively obvious that size alone would not be the only factor affecting effort. In addition, by explaining more of the effort variation using other factors, we might be able to better identify the most plausible relationship between size and effort.

Modeling the Impact of Team Size

The first variable we investigate is team size, i.e., defined here as peak staff load for the whole project. The importance of team size was considered in previous modeling efforts, such as the COCOMO model, Putnam's model [20], the Walston Felix Model [24], and Basili and Freburger [4]. But in most cases, team size was not identified as a significant factor on productivity assessment (see [8]). Conte et al. [8] suggested that team size may have been captured indirectly by other factors correlated with team size. Conte et al. then derived a productivity model including team size. This model depends on an assumed average degree of interaction among the developers and is based on the observation that, as the team size grows, the number of communication paths will also tend to grow.

Model Specification	Paramater Estimates		Std Error	R^2 / R^2 adj	MMRE	Pred(.25)	Obs
$Effort = a + (b \times KLOC) + (c \times TEAM)$	a	-11480.73	3698.86	0.72 / 0.71	0.73	20%	54
	b	142.21	43.77				
	c	3195.69	413.62				
$\ln(Effort) = a + (b \times \ln(KLOC)) + (c \times \ln(TEAM))$	a	6.23	0.22	0.72 / 0.76	0.48	31%	54
	b	0.39	0.08				
	c	0.99	0.13				

Table 5: Linear and log-linear model: effort vs. system size and team size

Simmons [22] investigated the influence of group communication and design partition on productivity. He

found that both design partition and communication are influential factors that may cause a ten-fold decrease in productivity. An effective design partition gets more important when staff size increases. He also reports a team size of eight people as being optimally efficient in his study.

In Table 5 we show the results of adding team size to our previous two plausible models[1]. This dramatically increases the R^2 value to approximately 0.72 in both cases. All parameters in both models are statistically significant.

Model Specification	Parameter Estimates		Std Error	R^2 / R^2 adj	MMRE	Pred(.25)	Obs
$Effort = a + (b \times KLOC) + (c \times TEAM) + (d \times Factor3)$	a	-5482.54	6295.33	0.77 / 0.75	0.85	30%	40
	b	205.87	49.39				
	c	3307.72	446.91				
	d	-3786.73	1850.397				
$\ln(Effort) = a + (b \times \ln(KLOC)) + (c \times \ln(TEAM)) + (d \times \ln(Factor1))$	a	1.30	1.34	0.79 / 0.78	0.41	40%	40
	b	0.44	0.08				
	c	0.74	0.14				
	d	1.84	0.48				

Table 6: Linear and Log-linear model: results from stepwise regression

The result for the log-linear model can be interpreted as follows: the higher the team size, the higher the impact of system size on effort, and vice-versa. For large systems, the impact of larger teams increases since more communication overhead is required. This is illustrated by Figure 1.

The results of performing the J-test are shown in Table 7. Given that we use an alpha level of 0.05 for statistical testing, these results indicate that including the Team Size variable does not help us identify the most plausible model specification. Again, both the linear and log-linear models are plausible, although the linear model is closer to be rejected.

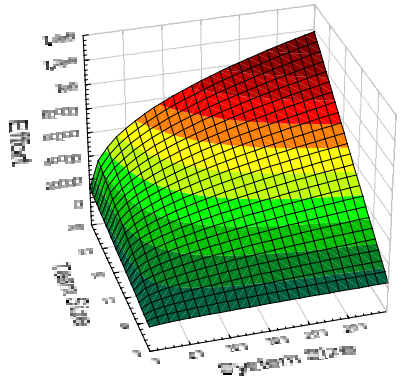

Figure 1: Relationship among Effort vs. Team Size and System Size

H_0	H_1	p-value for λ
Linear model	Log-linear model	0.0938
Log-linear model	Linear model	0.8423

Table 7: J-test results for models from Table 5

Modeling the Impact of COCOMO factors

The next set of variables that we consider are the COCOMO factors. It has been demonstrated through a number of previous studies that the COCOMO productivity

[1] Note that the reduced sample size is due to missing data in the Team Size variable. We do not have any reason to believe that there is a systematic bias in the provision of Team Size data (i.e., that there is a relationship between whether respondents provide this data and actual team size).

factors are not independent of each other [16][19]. In the ESA Space and Defense database seven of the COCOMO factors are collected. We therefore performed a principal components analysis on these seven variables to determine which are the underlying concepts. The results are shown in Table 8.

These results clearly indicate three distinguishable concepts (factors). The first factor relates to the constraints usually imposed on embedded, real-time systems (high reliability requirements, high storage and timing constraints). The second concept concerns the use of modern software engineering practices and powerful tools, which usually come together. The third factor captures the knowledge about the development platform (i.e., virtual machine) and the programming language. It is expected that if the platform is volatile, then it is unlikely that there will be sufficient up-to-date knowledge about it and the programming language in use on this platform, with its programming support tools.

Each of the principal components can be utilized as a single variable that is entered into the regression models that we are building. Each of them can be seen as a weighted sum of the most important variables (high factor loadings) to produce a composite variable. In our case, for example, Factor 1 includes variables RELY, TIME, and STOR.

	Factor 1 (RTC: Real Time Constraints)	Factor 2 (SEP: Software Engineering Practices)	Factor 3 (EXP: Experience)
VIRT	0.22	-0.3	**-0.67**
RELY	**0.83**	-0.22	0.004
TIME	**0.89**	0.25	0.007
STOR	**0.73**	0.43	-0.004
MODP	-0.13	**-0.68**	-0.42
TOOL	-0.09	**-0.85**	0.15
LEXP	0.17	-0.21	**0.84**

Table 8: Results of Principal Components Analysis on the seven COCOMO factors (73% of variation explained). Rotated components.

The result of constructing a log-linear model and a linear model including the team size and the three factors mentioned above are summarized in Table 6. Note that a backward stepwise regression procedure was followed. Compared to the models in Table 5, the R^2 values increased, but not dramatically.

The results in Table 6 can be interpreted as follows. For the linear model, we see that there is a linear relationship between experience and effort. The relationship is negative because of the way this principal component was coded: negative values indicate lack of experience. For the log-linear model, the results indicate that as real-time constraints (Factor 1) increase, there is an increase in effort. For greater real-time constrained projects, the impact of team increases. For large team sizes, the impact of Factor 1 increases and effort grows even faster than Factor 1 following a convex curve, as illustrated in Figure 2.

One potential explanation is that for a fixed team size, there is a ceiling effect on the amount of effort that can be consumed on projects. In addition, as team size increases, more inspections and integration testing may be needed to meet a given level of reliability and performance.

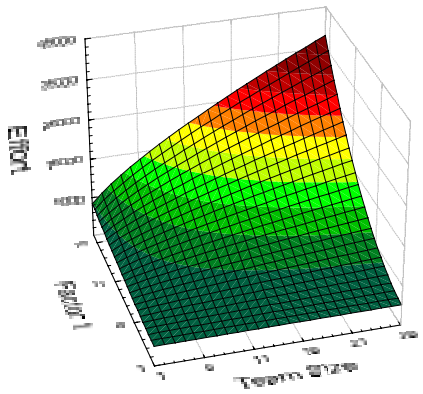

Figure 2: Relationship Effort vs. Factor 1 and Team Size

The results of the J-test for comparing these two models are shown in Table 9. These results clearly indicate the superiority of the log-linear model to the linear model, since H_0 is rejected for the linear model. Therefore, by adding new variables to further explain the variation in effort, we have found that a log-linear model provides the most plausible explanation of effort. This model explains 79% of the variation in effort**.** Furthermore, this model shows economies of scale as the coefficient *b* in the log-linear model is lower than 1 and significantly different from 1 (Table 6).

By adding new factors in the model, we are able to better distinguish the most plausible functional form. Our results might explain why it was difficult, in many studies, to differentiate the most plausible functional form when looking only at the size/effort relationship.

H_0	**H_1**	**p-value for λ**
Linear model	Log-linear model	0.0002
Log-linear model	Linear model	0.2086

Table 9: J-test results for models from Table 8

Since multicollinearity can have a significant impact on multivariate coefficients, we also performed a multicollinearity analysis. We wanted to ensure that we could interpret the regression coefficients presented above

(Table 6). For each model, we built regression equations using in turn each regressor as a dependent variable and the others as independent variables. We calculated the variance inflation factor (VIF) for each of the regression equations, as described in [21]. Multicollinearity is declared to exist whithin a model, if any VIF is greater or equal to a certain threshold, usually 10. In our case, all of the VIF's were very low and below 10. Thus, we can conclude that multicollinearity is very low in our multivariate models.

Another type of model that we did test during our study was one including interaction effects in both, the linear and log-linear model. For the log-linear model, it consists of the addition of a multiplicative term including two logarithms. This kind of model is based on the hypothesis that other variables (e.g., Team Size) have an impact on the extent of economies of scale since they would affect the KLOC exponent value. We did not find the interaction terms in both models to be statistically significant, indicating that there is no interaction effect. Therefore, with respect to the log-linear model, we do not have evidence that the extent of economies of scale is affected by other contextual factors.

5 CONCLUSIONS

Our goal in this paper was to investigate a few of the important questions regarding software cost modeling by using a part of the European Space Agency project database. We looked at the plausibility of various functional forms modeling the effort/ product size relationship, the extent of economies of scale, the impact of factors such as team size and various COCOMO factors.

Our results provide a log-linear model that explains a large proportion of the variation in project effort. This result is surprising considering that we are using a multi-organization database and would expect more inconsistency in the data. system size and team size are the factors having the largest impact on the goodness of fit of the model.

When trying to identify the most plausible relationship between effort and size, two functional forms are equally plausible: linear and log-linear. However, when integrating team size and COCOMO factors as covariates, thus explaining some of the effort variance not explained by system size, the log-linear model appears to be the most plausible one. On the other hand, no quadratic term was found significant when added to these two models so that the models referred to as quadratic and translog [1] are not plausible based on this data set.

The relationship between effort and size is difficult to identify and characterize because it may be blurred by other factors. This may explain the inconsistency of results in the literature regarding the nature and form of the effort / size relationship.

Another important result was that team size has a very substantial impact on project productivity, thereby confirming that compressing cycle time, which results into larger teams, comes at a substantial additional cost. Such a result was suggested by several authors in the past [7][8] and is confirmed here in quantitative terms, both for the linear and log-linear models.

Our database contained data for seven of the COCOMO factors that were deemed more important by the European Space Agency. Similarly to other studies [16][19], we have identified numerous interrelationships between these factors. In fact, a principal component analysis reveals that the seven factors capture three concepts: (1) the typical features of embedded systems such as high real-time and storage constraints, and high reliability requirements, (2) the use of modern programming practices and tools, (3) the working knowledge of the programming language and development platform. When we tried to use this principal components to improve the effort model equations, (1) and (3) appeared significant in the linear and log-linear models, respectively. One of the reasons why the two factors do not appear in both models is that, at this point, only 40 observations remain in the sample and cannot allow much more that 4 estimated parameters. At any rate, although significant, the selected COCOMO principal components do not have a substantial effect on the goodness of fit of the models. The investigation of additional factors explaining more variance in the ESA data set is therefore an important issue.

We can only confidentially claim that these results are applicable to European space and military projects. However, since we have a rather representative, recent, and large database, we believe we can make a number of practical recommendations. First, for the particular application domains under study, a log-linear relationship between effort and size should always be investigated. If different model specifications between effort and system size are tested and compared, then other covariates should be included in the model to better differentiate the available specification alternatives. We have seen in this study that it can make a significant difference in the analysis output when, for example, using the J-test. This may explain why so many studies on this topic yield contradictory results. Another practical recommendation is that team size should be measured in some way and considered in the prediction models. Many existing databases do not, however, consider this cost driver in an explicit manner.

ACKNOWLEDGEMENTS

We wish to thank the European Space Agency and INSEAD for giving us access to the ESA data. In particular, we are grateful to Benjamin Schreiber from the ESA/ESTEC center in the Netherlands. The ESA database is accessible to any organization willing to contribute to the database with project data fulfilling a number of criteria, e.g., more than 12 months of effort. The data provided are then sanitized and made available in the next version of the

database.

We would also like to thank Arthur Harutyunyan for his help on the data analysis reported in this paper.

REFERENCES

1. Banker, R., Kemerer, C. Scale Economies in New Software Development. *IEEE Transactions on Software Engineering, vol. 15, no. 10* (1989), 1199-1205.
2. Banker, R., Kemerer, C. The Evidence on Economies of Scale in Software Development. *Information and Technology, vol. 36, no. 5* (1994), 275-282.
3. Bailey, J.W., Basili V.R. A Meta-Model for Software Development Resource Expenditures. In: *Proceedings of the 7th International Conference on Software Engineering* (1981).
4. Basili, V.R., Freburger, K. Programming Measurement and Estimation in the Software Engineering Laboratory. *Journal of Systems and Software, no. 2* (1981), 47-57.
5. Boehm, B., Software Engineering Economics. *Prentice Hall* (1981).
6. Boehm, B., Clark B., Horowitz, E., Westland, C. Cost models for future software life cycle processes: COCOMO 2.0. *Annals of Software Engineering, 1* (1995), 57-94.
7. Brooks, F.P. The Mythical Man Month. *Addision-Wesley Publishing Company* (1975).
8. Conte, S., Dunsmore, H., Shen, V. Software Engineering Metrics and Models. *Benjamin/ Cummings, Menlo Park CA,* (1986)
9. Cook, R.D. Detection of Influential Observations in Linear Regression. *Technometrics, vol. 19* (1977), 15-18.
10. Davidson, R., McKinnon, J., Several Tests for Model Specification in the Presence of Alternative Hypotheses. *Econometrica, vol. 49, no. 3* (1981), 781-93.
11. Dunteman, G.,H.. Principal Component Analysis. *Sage University paper series on Quantitative Applications in the Social Sciences* (1989).
12. Hu, Q. Evaluating Alternative Software Production Functions. *IEEE Transactions on Software Engineering, vol. 23, no. 6* (1997), 379-87.
13. Kemerer, C.F. An Empirical Validation of Software Cost Estimation. *Communications of the ACM, vol. 30, no. 5* (1987), 417-29.
14. Jones, C. Programming Productivity. *New York, McGraw-Hill* (1986).
15. Johnston, J., DiNardo, J. Econometric Methods. *New York: McGraw-Hill* (1997).
16. Kitchenham, B. Software Development Cost Models. *In: Software Reliability Handbook, P. Rook (Ed.) Elsevier Applied Science, New York* (1990).
17. Kitchenham, B. Empirical Studies of Assumptions That Underlie Software Cost-Estimation Models. *Information and Software Technology, vol. 34, no. 4* (1992), 211-18.
18. Matson, J.E. Barrett, B.E., Mellichamp, J.M. Software Development Cost Estimation Using Function Points. *IEEE Transactions on Software Engineering, vol. 20, no. 4* (1994), 275-87.
19. Maxwell, K., Van Wassenhove, L., Dutta, S. Software Development Productivity of European Space, Military, and Industrial Applications. *IEEE Transactions on Software Engineering, vol. 22, no. 10* (1996), 706-18.
20. Putnam, L.H. A General Empirical Solution to the Macro Software Sizing and Estimating Problem. *IEEE Transactions on Software Engineering, vol. SE-4, no. 4,* (1978), 345-61.
21. Ryan, T.P. Modern Regression Methods. *John Wiley & Sons, Inc., New York* (1997).
22. Simmons, D.B. Communications: a software group productivity dominator, *Software Engineering Journal, November,* (1991), 454-462.
23. Subramanian, G.H., Breslawski, S. Effort Estimation Dimensionality Reduction. *Journal of Systems and Software, vol. 21* (1993), 187-196.
24. Walston C.E., Felix C.P. A method of programming measurement and estimation. *IBM Systems Journal, vol. 16,. no. 1* (1977), 54-73.