Fraunhofer Einrichtung Experimentelles Software Engineering

Explaining the Cost of European Space and Military Projects

Authors
Lionel C. Briand
Khaled El Emam
Isabella Wieczorek

IESE-Report No. 030.98/E
Version 1.0
May 1998

A publication by Fraunhofer IESE

Fraunhofer IESE is an institute of the Fraunhofer Gesellschaft.
The institute transfers innovative software development techniques, methods and tools into industrial practice, assists companies in building software competencies customized to their needs, and helps them to establish a competetive market position.

Fraunhofer IESE is directed by
Prof. Dr. Dieter Rombach
Sauerwiesen 6
D-67661 Kaiserslautern

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

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. The only way to address these issues and obtain a generalizable conclusion is to investigate them on as many data sets as possible. 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 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 sets. 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.

1 Introduction

Many studies have looked at the relationship between product size, development effort, and other factors [3][4][5][10][11][15][16][18]. One important line of research has aimed to (1) determine whether there were (dis)economies of scale, (2) the exact nature of the functional form of the effort/size relationship (linear, quadratic, exponential (or log-linear [1][2]), translog), and (3) the independence and role of COCOMO factors. The results and conclusions of these analyses vary a great deal due in part to the use of different datasets but also to different analysis techniques.

There have been a series of studies that investigates whether (dis)economies of scale exist in software production [1][2][15], and the functional form of the relationship between effort and size [10]. However, taken as a whole, these studies provide an inconsistent picture, making it difficult to draw general conclusions on whether there indeed are any (dis)economies of scale, and what the functional form of this relationship should be.

Better understanding of the relationship between effort and size is important to allow us to build cost prediction models which have correct specifications and to determine when to expect economies or diseconomies of scale. However, there is no a priori reason to believe that such relationships should be constant across environments since different organizations may handle large projects with varying degrees of success and show different fixed project overheads. This may in part explain the variability in results shown by existing studies. In addition, better understanding of how the COCOMO factors impact cost, and which ones are more likely to have a strong impact on cost, can help us determine what type of project information should be collected with great care and precision, from a cost modeling perspective.

In this paper, we focus on identifying the most plausible functional forms between effort and size, and then continue by determining how COCOMO factors actually relate to effort and interact with product size measured as KLOC. Not all COCOMO factors are collected here but only a subset of seven of them is present in the data set. Others, such as team size, are collected in a different way, e.g., by counting the maximum team size at any time during development. We use a large, multi-organization database which contains projects in the space and defense domains, coming from a variety of European countries and contracted by the European Space Agency (ESA). Since this database covers a large number of organizations, we hope to be able to better generalize our findings. On the other hand, by considering one application domain, we hope to have a reasonable degree of consistency in our data.

In the database investigated, the relationship between effort and size appears to be either linear or log-linear (i.e., $a.KLOC^b$) since both functional forms appear plausible, as discussed below. There is no evidence of a "most productive scale size", as suggested by Banker and Kemerer [1]. Although they show a similar R^2, the mean magnitude of relative error (MMRE) of the log-linear model is lower, which is to be expected since, as the result of the logarithmic transformation, it grants a lower weight to smaller projects that usually have a large impact on MMRE. If this is an important criterion to the user of the model, then the log-linear model is likely to be more useful. It was also found that, like in other studies [5][12], higher team size results in lower productivity due in part to larger communication overhead. Such results are consistent with the underlying assumptions of the COCOMO and Putnam models. However, the impact of team size observed on this data set is very substantial and suggesting that such a measure should be systematically and carefully collected. Finally, some 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.

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 interretationships 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 5 concludes the paper and and gives some information about our future research directions.

2 Background

2.1 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 [4]. 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, grows at a faster rate than project size [12], 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 [5], the potential for personality conflicts [4], and more complex system interfaces [6].

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 [4].

Kitchenham [15] 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 [13] 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 [10] revisited some of the datasets already investigated and concluded that, over 9 datasets, 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.

A promising line of enquiry was instigated by Hu’s [10] work since this provides an objective and statistically valid general basis 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 [11] there is one extreme outlier that has a substantial influence on the results of regression analysis (see [16]). 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 questionnable since there would be inconsistencies in the manner in which both effort and size (in Lines of Code) are measured across these data sets. Fourth, 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”. Finally, the possibility that there really is not a universal law that defines the relationship between effort and size was not considered as an interpretation of the inconsistent results obtained during the analysis of eight different data sets.

Clearly then, it is imperative to continue studying the functional form of the relationship between effort and size. An investigation should be open to the possibility that, even within a single data set, there may be subpopulations of projects that exhibit different laws between size and effort.

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.

2.2 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 [14] shows that there is a relationship between two of the COCOMO factors based on the COCOMO dataset itself. Similarly, through principal component analysis, Kitchenham [15] found that, out of 21 cost factors, seven principal components accounted for 75% of the effort variability in a data set of 28 projects. Maxwell et. al. [17] analysed 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). Furthermore, they found that the two factors *modern programming languages* and *tool usage* were highly correlated. Thus, they did not include them together when building the multivariate regression models. It is not clear why the grouped factors (i.e., principal components) were not used, instead of the individual cost factors, in order to build the multivariate models.

The studies presented above suggest that COCOMO factors do not capture independent dimensions. In the remainder of this paper, we will therefore perform principal component analysis in order to identify the underlying dimensions captured by the COCOMO factors in our dataset 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.

3.1 Data Source

The database used in this study is the European Space Agency (ESA) multi-organisation 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 comparabiliy of the responses. Each data supplier regularly receives data analysis reports of the dataset.

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.

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	interval	1 KLOC=1000 LOC
EFFORT	Effort for SW project	interval	Person hours , where 144 person hours=1 person month
TEAM	Maximal team size on one stage of a project	interval	
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 constriant	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

3.2 Comparison of Models

3.2.1 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.

3.2.2 Comparing the models' goodness of fit

The two traditional ways of assessing and comparing the goodness of fit of cost models is to compute their coefficient of determination R^2 (adjusted for the number of variables) and their mean magnitude of relative error (MMRE) [6]. 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 since, because of the MRE mathematical formulation, smaller projects tend to yield the larger MREs. For example, the log-linear model, because of its logarithmic transformation, grants more weight to smaller projects than the linear model and, therefore, its 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 comparison 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 [10], non-nested models cannot be compared using the R^2 since this one is affected by the use of different

3.2.5 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

Model Specification	Paramater Estimates		Std Error	R^2 / R^2_{adj}	MMRE	Obs
$Effort = a + (b \times KLOC)$	a (not sign.)	6447.69	3364.5	0.41 / 0.40	1.24	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	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	64
	b	0.73	0.082			
$\ln(Effort) = a + (b \times \ln(KLOC)) + c \times (\ln(KLOC))^2$	a	7.38	0.54	0.42 / 0.41	0.67	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

In the literature there have been discussions 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.

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.

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

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

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.

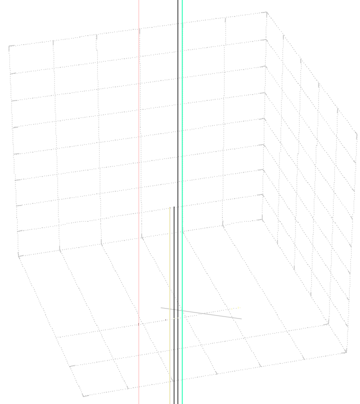
4.1.2 Modeling the Impact of Team Size

The first variable we investigate is Team Size. It is clear that the larger the team size the more effort will be spent. The importance of Team Size was considered in previous modelling efforts, such as the COCOMO model [4] and Putnam's model [18]. In Table 5 we show the results of adding team size to the 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	Paramater Estimates		Std Error	R^2 / R^2_{adj}	MMRE	Obs
$Effort = a + (b \times KLOC) + (c \times TEAM)$	a	-11480.73	3698.86	0.72 / 0.71	0.73	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 / 0.715	0.48	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

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.



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

Table 6: J-test results for models from Table 5

4.1.3 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 factors are not independent of each other [14][17]. In the ESA Space and Defense data base seven of the COCOMO factors are collected. We therefore performed a principal components analysis on these seven variables to determine which are the underlying dimensions. The results are shown in Table 7.

These results clearly indicate three dimensions. The first dimension relates to the constraints usually imposed on embedded, real-time systems (high reliability requirements, high storage and timing constraints). The second dimension 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.

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

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.

The result of constructing a log-linear model and a linear model including the team size and the three dimensions mentioned above are summarized in Table 8. Note that a backward stepwise regression procedure was followed. Compared to the models in Table 6, the R^2 values increased, but not dramatically.

The results in Table 8 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.

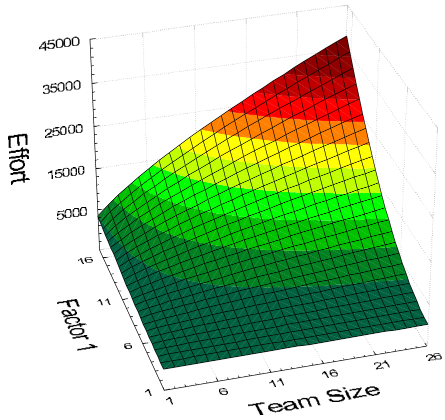


Figure 2: Relationship Effort vs. Factor 1 and Team Size

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.

Model Specification	Parameter Estimates		Std Error	R^2 / R^2_{adj}	MMRE	Obs
$Effort = a + (b \times KLOC) + (c \times TEAM) + (d \times Factor3)$	a	-5482.54	6295.33	0.77 / 0.75	0.85	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
	b	0.44	0.08			
	c	0.74	0.14			
	d	1.84	0.48			

Table 8: Linear and Log-linear model: results from stepwise regression

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 (In Table 8, coefficient b is significantly different from 1).

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 α
Linear	Log-linear	0.0002
Loglinear	Linear	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 5 and Table 8). 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 [19]. 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.

4.2 Discussion

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. System size and team size are the factors having the largest impact on the goodness of fit of the model. COCOMO factors, although significant, play a minor role with that respect.

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, the log-linear model appear to be the most plausible.

We can only confidentially claim that these results are applicable to European space and military projects. However, they do raise a number of issues on the investigation of economies of scale and the development of effort 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 Conclusion

Our goal 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 showed that both a linear and a log-linear relationship is plausible, when not taking into account other cost factors. However, after including team size and some COCOMO factors in the models, the most plausible functional form is the log-linear 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. These results are likely to suggest that our sample of projects comes, in somewhat equal proportions, from two populations for which we have, respectively, a linear and log-linear relationship between effort and product size.

As discussed in the introduction, we should expect that some organizations are better prepared to deal with large projects and therefore exhibit economies of scale in their projects, whereas others do not or even show diseconomies of scale. There is no reason to believe there is only one, universal, plausible relationship, as it has often been discussed in the literature. The emphasis of our research should rather be on identifying conditions that are favorable to economies of scale.

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 [5][6] and is confirmed here in quantitative terms, both for the linear and log-linear models. Many cost databases do not capture (or at least not precisely) this parameter and our result suggests that its impact should be closely monitored in future studies.

Our database contained data for seven of the COCOMO factors that were deemed more important by the European Space Agency. Similarly to other studies [14][17], we have identified numerous interrelationships between these factors. In fact, a principal component analysis reveals that the seven factors capture three dimensions: (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 remains 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.

6. Acknowledgements

We wish to thank the European Space Agency and INSEAD for giving us access to the data. 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.

7. References

[1] Banker, R., Kemerer, C., Scale Economies in New Software Development, IEEE Transactions on Software Engineering, vol.15, no. 10, pp. 1199-1205, 1989

[2] Banker, R., Kemerer, C., The Evidence on Economies of Scale in Software Development, Information and Technology, vol.36, no. 5, pp. 275-282, 1994.

[3] Bailey, J. W., Basili V. R., A Meta-Model for Software Development Resource Expenditures, Proc. International Conference on Software Engineering, IEEE Computer Society Press, 1981

[4] Boehm, B., Software Engineering Economics, Prentice Hall, 1981

[5] Brooks, F. P., The Mythical Man Month, Addision-Wesley Publishing Company, 1975

[6] Conte, S., Dunsmore, H., Shen, V., Software Engineering Metrics, and Models, Benjamin/Cummings, Menlo Parc CA, 1986

[7] Cook, R.D., Detection of Influential Observations in Linear Regression, Technometrics, vol.19, pp.15-18, 1977

[8] Davidson, R., McKinnon, J., Several Tests for Model Specification in the Presence of Alternative Hypotheses, Econometrica, vol. 49, no. 3, pp. 781-93, 1981.

[9] Dunteman, G., H., Principal Component Analysis, Sage University paper series on Quantitative Applications in the Social Sciences, 1989

[10] Hu, Q., Evaluating Alternative Software Production Functions, IEEE Transactions on Software Engineering, vol. 23, no. 6, pp. 379-87, 1997.

[11] Kemerer, C., F., An Empirical Validation of Software Cost Estimation, Communications of the ACM, vol. 30, no. 5, pp. 417-29, 1987

[12] Jones, C., Programming Productivity, New York, McGraw-Hill, 1986

[13] Johnston, J., DiNardo, J., Econometric Methods, New York: McGraw-Hill, 1997

[14] Kitchenham, B., Software Development Cost Models, In: Software Reliability Handbook, P. Rook, Editor, Elsevier Applied Science, New York, 1990

[15] Kitchenham, B., Empirical Studies of Assumptions That Underlie Software Cost-Estimation Models, Information and Software Technology, vol. 34, no. 4, pp. 211-18, 1992

[16] 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, pp. 275-87, 1994

[17] 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, pp. 706-18, 1996

[18] Putnam, L. H., A General Emperical Solution to the Macro Software Sizing and Estimating Problem, IEEE Transactions on Software Engineering, vol. SE-4, no. 4, pp. 345-61, 1978.

[19] Ryan, T.P., Modern Regression Methods, John Wiley & Sons, Inc., New York, 1997

Document Information

Title:	Explaining the Cost of European Space and Military Projects
Date:	May 1998
Report:	IESE-030.98/E
Status:	Final
Distribution:	Public
	also published as ISERN-98-19