SOFTWARE DEVELOPMENT EFFORT ESTIMATION BASED ON SIGNIFICANT PRODUCTIVITY FACTORS: *GENERAL AND COMPANY SPECIFIC MODELS*

by

K. MAXWELL*
L. N. VAN WASSENHOVE**
and
S. DUTTA†

95/80/TM

* Reseach Associate, at INSEAD, Boulevard de Constance, 77305 Fontainebleau Cedex, France.

** Professor of Operations Management and Operations Research, at INSEAD, Boulevard de Constance, 77305 Fontainebleau Cedex, France.

† Associate Professor of Information Systems, at INSEAD, Boulevard de Constance, 77305 Fontainebleau Cedex, France.

Financial support from the European Space Agency (ESA) is gratefully acknowledged.

Software Development Effort Estimation

based on

Significant Productivity Factors:

General and Company Specific Models

Katrina Maxwell, Luk Van Wassenhove and Soumitra Dutta

INSEAD
Boulevard de Constance
77305 Fontainebleau Cedex, France
(33-1) 6072-4266

Abstract

The limitations of existing software development effort estimation models suggest the need for the development of simple effort estimation models based on the prior determination of the independent factors which explain the productivity variation of a given database. This paper builds on our previous research which investigated the software productivity of a European Space Agency database consisting of 108 software development projects. The objectives of this paper were firstly, to develop and evaluate simple empirical effort estimation models which include only those productivity factors found to be significant for these projects; and secondly, to determine if models based on a multi-company database can be successfully used to make effort estimations within a specific company. This was accomplished by developing company specific effort estimation models based on the significant productivity factors of a particular company and by comparing the results with those of the general ESA model. To our knowledge, no other published research has yet developed and analysed software development effort estimation models in this way. Our study found that the best general effort estimation models were based on the size of the project and the main factors found to effect the productivity of the ESA dataset: application category, language, required software reliability, main storage constraint and the use of modern programming practices or software tools. The best company specific model was based on the two factors which explained the productivity of the individual company. Effort predictions made for the individual company's projects using the general models were a great deal less accurate than the company specific model. However, it is likely that in the absence of enough resources and data for a company to develop its own model, the application of general models may be more accurate than the use of guessing and intuition.

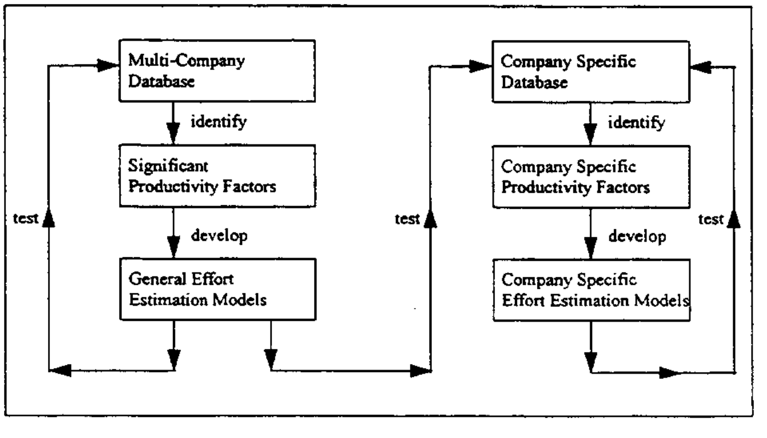
Introduction

Accurate effort and cost estimation of software applications continues to be a critical issue for software project managers. In a recent survey, Lederer and Prasad (1992) found that two-thirds of all major software projects substantially overran their estimates. Although more than 20 software cost estimation models were already in existence as early as 1981 (Mohanty 1981), Lederer and Prasad found that, in practice, software development cost estimators still rely more heavily on their personal memory of the cost of similar projects than on other estimating processes. This technique was not associated with accurate estimates. In addition, the use of guessing and intuition were found to be highly positively correlated with the percentage of large projects overrunning their estimates. Only the use of similar, past projects based on documented facts, the use of a simple arithmetic formula, and the use of established standards were found to lead to greater accuracy in software cost/effort estimates.

Over the past years, many effort estimation models have been developed (Aron 1976, Walston and Felix 1977, Putnam 1978, Bailey and Basili 1981, Basili and Freburger 1981, Boehm 1981, Albrecht and Gaffney 1983, Conte et al 1986, Kemerer 1987, Matson et al 1994, to name a few); however, because of differences in data collected, project types and environmental factors among software development sites, these models are generally only valid within the organization in which they were developed (Bailey and Basili 1981). One major obstacle to the transportability of these models appears to be a lack of understanding of the factors explaining the differences in productivity among projects. As there exist literally hundreds of parameters which can affect software development effort (DeMarco 1982), and as only a few of these may effect the productivity within a given environment (Banker et al 1991, Kitchenham 1992, Mukhopadhyay and Kekre 1992), it is important to analyse the accuracy of effort estimation models based on the prior determination of the factors found to explain the productivity of projects in a given database. Furthermore, it is also important to determine if effort estimation models based on the significant productivity factors of a multi-company database can be successfully used to make effort estimations within a specific company. To our knowledge, no other published research has yet developed and analysed software development effort estimation models in this way.

In this paper we present the results of our effort estimation analysis of the European Space Agency software development database which, at the time of the analysis, consisted of 108 projects from 37 companies in 8 European countries. This database is unique in that we have found no other published research which analyses such a large number of European non-MIS applications. This paper seeks to build on our previous research which investigated the productivity differences of European space, military and industrial applications (Maxwell et al 1994) by developing simple empirical effort estimation models which include only those productivity factors found to be significant in these application areas. As we previously determined that organizational differences account for most of the productivity variation of projects in the ESA dataset, it was decided to develop general ESA effort estimation models with the data from one company removed, and then to test the general ESA models on the removed company. We also develop simple company specific effort estimation models based only on the factors found to effect the productivity of this particular company. The results of the best company specific models are then compared to the models developed using the ESA dataset. This is illustrated in Figure 1.

Figure 1.Effort Estimation Model Development and Testing on Databases



In our analysis, parsimonious models are employed to examine the impact of differences in company, country, language, category, environment, start year, team size, project duration and system size, as well as the following seven COCOMO factors (Boehm

1981): required software reliability, execution time constraint, main storage constraint, virtual machine volatility, programming language experience, use of modern programming practices and use of software tools, on productivity and effort estimation.

The remainder of the paper is organized as follows. An overview of prior effort estimation research is presented, followed by a description of the database and our effort estimation analysis. The results of the effort estimation analysis are then presented. A summary of the results can be found in the concluding section.

Prior Research

Although there exists some recent research on the use of case-based reasoning models for software effort estimation (Mukhopadhyay et al 1992), effort estimation research has largely focused on the study of algorithmic techniques. These techniques can be categorized by the type of formula used to calculate total effort: single or multivariable, static or dynamic with regard to staffing, based on historical data or theoretical assumptions, and can use a top-down or bottom-up approach (Basili 1980). Heemstra (1992) provides a detailed overview of some recent software effort estimation models. An excellent summary of the effort estimation models available prior to 1981 can be found in Mohanty (1981).

Most effort estimation models consist of two phases: in the first phase, an estimate of the software size is made; and in the second phase, the effort and duration of the project are estimated based on the estimated software size. As many factors besides software size have an influence on the effort and time needed to develop software, most effort estimation models also include an adjustment for a number of productivity factors (Kitchenham 1992). Models based on function point analysis (Albrecht and Gaffney 1983) are focused more on the sizing stage while others, such as COCOMO (Boehm 1981), are focused more on the productivity stage (Heemstra 1992).

In a study by Mohanty (1981), the cost of one software development project was found to vary by nearly 800% when estimated using 12 different cost/effort estimation models. As the size of the project and cost per instruction were kept constant, he concluded that the primary cause of this variation was environmental. This is because each model was developed using a specific database from a given company environment and thus embodied the productivity factors, work patterns, and management practices of that company's environment.

Some of the many factors that appear to influence software development productivity are: people factors, such as experience and team size; process factors, such as programming language and tools; product factors, such as complexity and software type; and computer factors, such as storage and timing constraints (Conte et al. 1986). The combination and interaction of all of these factors makes effort estimation very difficult. An overview of some productivity factors considered by past researchers can be found in Tables 1 and 2.

[Tables 1 and 2 here]

In the past, many researchers have based their models on a large number of productivity factors. In a IBM study by Walston and Felix (1977), 29 factors that were significantly correlated with productivity were found. In an analysis of data from the NASA/Goddard Space Flight Center, Bailey and Basili (1981) identified 21 productivity parameters . At ITT, Vosburgh et al. (1984) found 14 significant productivity factors. In Boehm's COCOMO model (1981), 15 software factors which had a significant impact on productivity were identified.

However, Kitchenham (1992), in her review of some of the assumptions built into conventional software cost/effort models, found no support for the assumption that a large number of productivity factors, or size adjustment factors for models based on function points, are necessary. Few adjustment factors were found to have a significant effect on productivity within one environment. Mukhopadhyay and Kekre (1992) obtained good results in their development of an application feature based method for the early effort estimation of process control applications using only two productivity factors. The results of Banker et al (1991) suggested that a relatively small number of critical factors may explain a large amount of productivity variation at a specific site.

In addition, in models such as COCOMO, the adjustment factors are treated as if they are independent even though some are not. In Kitchenham's (1992) analysis of the COCOMO dataset, she found that some of the 15 adjustment factors used did not even significantly explain the productivity of the dataset used to develop the model. An evaluation of four algorithmic models used to estimate software effort (SLIM, COCOMO, Function Points and ESTIMACS) undertaken by Kemerer (1987) also determined that none of the models tested sufficiently reflected the underlying factors affecting productivity. Thus strong empirical evidence suggests the need for the development of a simple effort estimation model based on the prior determination of a small number of independent factors which explain the productivity variation of a given database.

The effort estimation models presented in this paper build on our recent study of the factors which influence the software development productivity of European space, military and industrial anplications (Maxwell et al 1994). In this study, we found that organizational differences accounted for most of the productivity variation of projects in the ESA dataset . Some of this variation was due to the differences in the application category and programming language of projects in each company. In addition, high productivity was found in those companies which undertook projects with low reliability requirements, low main storage constraints, low execution time constraints and which had a high use of tools and modern programming practices. We also determined that productivity decreased with increasing team size and project duration, and that programming language experience had no significant effect on productivity.

The ESA Database

In 1988, the European Space Agency (ESA), faced with the evaluation of proposals for large space programmes and their huge needs for software development, began compiling a software metrics database focusing on cost/effort estimation and productivity measures. Subsequently, the database was expanded to include military and industrial applications. This activity is ongoing, and at the time of this analysis contained a selection of 108 completed projects from 37 companies in 8 European countries. The database collection effort consists of contacting each supplier of data on a regular basis to determine if suitable projects are nearing completion. When a completed questionnaire is received, each supplier of data is telephoned to ensure the validity and comparability of his responses. In return, each data supplier receives periodic data analysis reports and diskettes of the sanitized dataset.

The 108 projects at the time of this analysis represented 5.51 million lines of source code (range: 2000-413000, average: 51010, median: 22300), 22 development languages or combinations of languages, and 30125 manmonths of effort (range: 7.8-4361, average: 284, median: 93). More details about the database are given in Figures 2a, 2b, 2c, 2d and Table 3.

Figure 2a.
Percentage of Projects by Main Application Environment

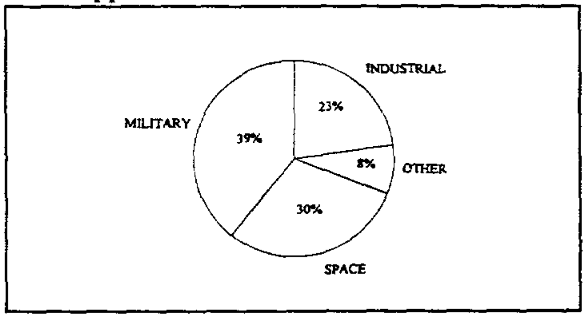


Figure 2b.
Percentage of Projects by Country

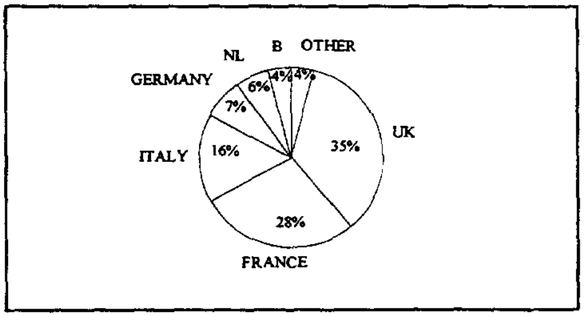


Figure 2c.
Percentage of Projects by Language

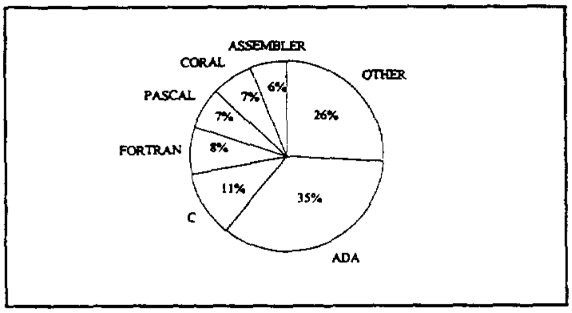


Figure 2d.
Percentage of Projects per Company
(Ten largest data suppliers)

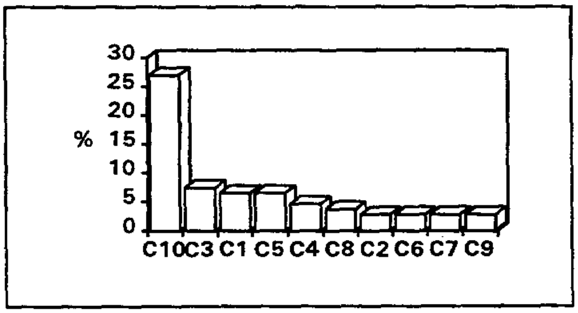


Table 3. Variables in ESA dataset

LANG		Application Programming Language
	ADA	Ada
	PAS	Pascal
	FOR	Fortran
	LTR	LTR
	C	C
	TAL	TAL
	COR	Coral
	AS	Assembler
ENV		Environment (Space, Military, Industry)
COMPANY		Company where project was developed
COUNTRY		Country where project was developed
CATEGORY		ESA Classification
	OB :	On Board
	MSG :	Message Switching
	RT :	Real Time
	GSE :	Ground Support Equipment
	SIM :	Simulators
	GRD :	Ground Control
	TL:	Tool
	OTH :	Other
TEAM		Maximum size of implementation team
DUR		Duration of project in months
KLOC		Kilo Lines of Code
YEAR		Start Year of Project
RELY		Required Software Reliability
TIME		Execution Time Constraint
STOR		Main Storage Constraint
VIRT		Virtual Machine Volatility
LEXP		Programming Language Experience
MODP		Use of Modern Programming Practices
TOOL		Use of Software Tools

In the ESA database, KLOC, Effort and Productivity are defined as follows:

KLOC: the amount of non-blank, non-commented delivered kilo lines of code. As the software developed in some projects consists of reused code, adaption adjustment factors (Boehm 81) were used to correct the size of the software. When no adaption adjustment factor was available, the new code size was used as a measure for the size.

Effort: The total effort was measured in manmonths and was defined as beginning at specifications delivery and ending at customer acceptance. The effort value covers all directly charged labour on the project for activities during this period. All effort data has been converted to manmonths of 144 manhours per manmonth.

Productivity: In software development terms, productivity is conventionally defined as source lines of code per manmonth (LOC/MM). It is a measure of the amount of product produced per unit of human effort. Although the lines-of-code metric is the subject of much debate, the fact remains that it is considered by many organizations as a more practical productivity metric than the currently available alternatives (Boehm 1987). In a recent study by Cusumano and Kemerer (1990), the lines-of-code metric was chosen to compare productivity across a number of organizations in the US and Japan. Previous research has also shown that function points and lines-of-code tend to be highly correlated in the case of new software development (Banker et al. 1991). Until the use of function and feature point methods become common for non-MIS applications, and particularly in the domain of space, military and industrial applications, statistical analysis undertaken of large heterogenous databases will have to rely upon measuring and analysing the productivity of these types of projects using lines-of-code metrics (Maxwell et al 1994).

Design of Analysis

In our recent study of the factors which influence the software development productivity of European space, military and industrial applications (Maxwell et al 1994), we found that organizational differences accounted for most of the productivity variation of projects in the ESA dataset. Therefore, in this study, we have analysed the data on two levels: multi-company and company specific, in order to identify and compare the productivity factors found at each level and to determine if effort estimation models developed using a multi-company database can be successfully used to make effort estimations within a specific company.

As nearly 27% of the data in the ESA database was supplied by a single company (see Figure 2d), it was decided to develop the general ESA effort estimation models with the data from this company removed in order to prevent the resulting models being overbiased, and then to test the general ESA models on the removed company. Thus the general ESA analysis was carried out on a subset of 79 projects. The company specific analysis was undertaken on the subset of 29 projects provided by company C10.

The analysis was performed in six phases.

Phase (1) The individual variables which explain the greatest amount of productivity variation of the ESA dataset after removal of the company C10 data were determined.

Phase (2) The best ESA productivity models were identified.

Phase (3) The best ESA effort estimation models based on significant productivity factors were selected.

Phase (4) The individual variables which explain the greatest amount of productivity variation of company C10 were determined.

Phase (5) The best company C10 productivity models were identified.

Phase (6) The best company C10 effort estimation models based on significant productivity factors were selected and compared with general ESA models.

Parsimonious models were employed to examine the impact of differences in company, country, language, category, environment, start year, team size, project duration and system size, as well as the following seven COCOMO factors (Boehm 1981): required software reliability, execution time constraint, main storage constraint, virtual machine volatility, programming language experience, use of modern programming practices and use of software tools, on productivity and effort estimation. As the data was not normally distributed, the measure of correlation used was Spearman's rank correlation coefficient (Fenton 1991). Any two variables with a correlation coefficient exceeding + or - .75 were considered to be highly correlated.

A General Linear Models procedure which can analyse the variance of unbalanced data was used for this analysis. Crossed effects of class variables were taken into consideration. The analysis was performed in an unbiased way: all values were considered as being equally reliable and relationships were extracted based on the face value of the data. Both additive and multiplicative (log) models were fit to the data.

Additive: $$Productivity = a + b \times x_1 + c \times x_2 + \cdots \quad (1)$$

$$Effort = a + b \times KLOC + c \times p_1 + d \times p_2 + \cdots \quad (2)$$

Multiplicative: $$Productivity = a \times x_1^b \times x_2^c \times \cdots \quad (3)$$

$$Effort = a \times KLOC^b \times p_1^c \times p_2^d \times \cdots \quad (4)$$

where a is a constant which varies with the significant class variables and p is a significant productivity factor.

The criteria for choosing the best estimation models were based on the accuracy and consistency of the effort estimates, as well as the simplicity and applicability of the model. Thus, a model based on few productivity factors was considered superior to a more complex model as long as there was not a substantial loss of accuracy and/or consistency. The accuracy of an estimation method was measured by the mean magnitude of relative error (MMRE) of the project estimates. A MMRE of 30% means that on average the estimates were within 30% of the actuals. The accuracy of an estimation method is inversely proportional to its MMRE. In addition, a second measure, called PRED(.25) or Prediction at Level .25, was used to measure the percentage of projects for which the predicted effort was within 25% of the actual effort. The consistency of the estimation method was measured by the correlation (CORR) between the estimates and the actuals. A consistent method should have a high positive correlation between actuals and estimates. For an in-depth discussion of accuracy and consistency refer to Mukhopadhyay and Kekre (1992).

Presentation of Results

Phase I - ESA Dataset Productivity Analysis Results of Individual Variables

The first phase of the analysis was concerned with determining which individual variables explain the greatest amount of productivity variation of the ESA dataset[1]. As it would not be wise to base our conclusions on the analysis of class levels that contain limited observations, analysis was undertaken on all of the data as well as for subsets that contain a sufficient number of observations at each class level. Models based on individual continuous variables were also

[1] In this paper, the ESA dataset refers to the ESA database with the data from company C10 removed

analysed and are used to explain the differences in the class variables. The summary of the analysis of significant individual class variables for the ESA database not including company C10 can be found in Figure 3. The level of significance is shown in parentheses. The single variable which explained the greatest amount of variance (68%) in the dataset was Company. This highlights the need for companies to establish their own software metrics database in addition to benchmarking their data against that of other companies.

Figure 3. Percentage of productivity variance explained by significant individual class variables in ESA dataset

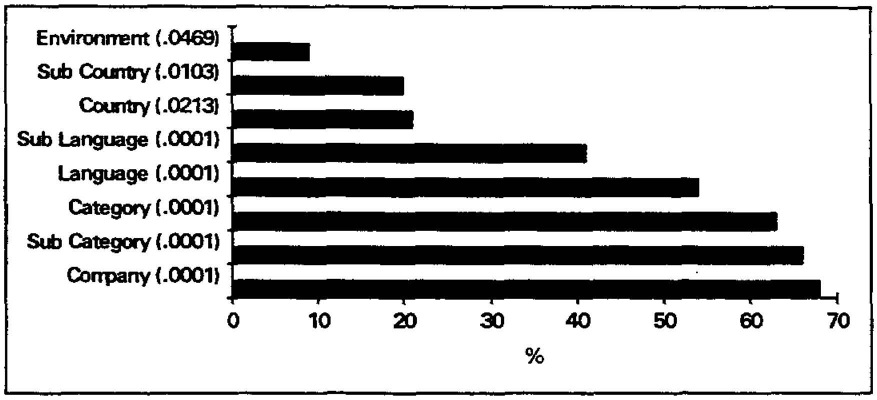
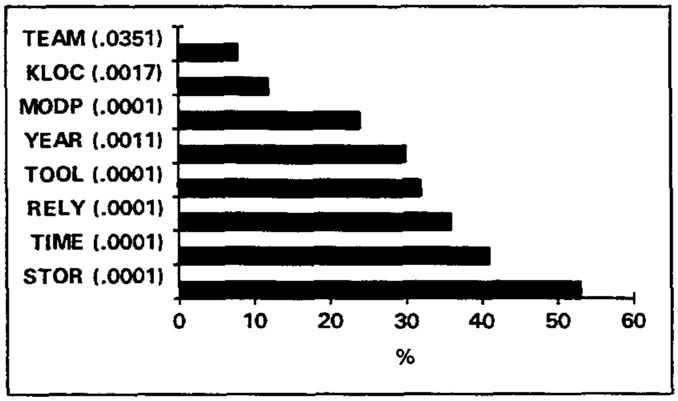


Figure 4. Percentage of productivity variance explained by significant individual continuous variables in ESA dataset



The summary of the productivity analysis of the significant individual continuous variables can be found in Figure 4. The level of significance is shown in parentheses. Productivity was found to significantly decrease with increasing team size, and to significantly increase over time (start YEAR of project) and increasing system size. Productivity was also found to significantly increase with increasing use of tools and modern programming practices, and with decreasing storage constraints, time constraints and required reliability. The average score of these five significant productivity factors for the 3 highest and the 3 lowest productivity software projects is shown in Figure 5. The productivity of these projects ranged from 39 to 1744 LOC/MM. The average scores of these productivity factors for the highest and lowest productivity companies can be found in Figure 6. The productivity of the companies ranged from 84 LOC/MM for company C4 to 853 LOC/MM for company C2. The average scores for the highest and lowest productivity categories is shown in Figure 7. The GSE category had an average productivity of 683 LOC/MM and the OB category had an average productivity of 132 LOC/MM. Figure 8 shows the averages scores for the highest and lowest productivity languages. Ada and Assembler had mean productivities of 548 and 63 LOC/MM respectively. No significant relationships were found between MODP, TOOL, STOR, TIME and RELY and the start YEAR of the project. This result is probably due to the effect of combining data from different companies.

Figure 5. Scores of productivity factors for highest and lowest productivity projects in ESA dataset

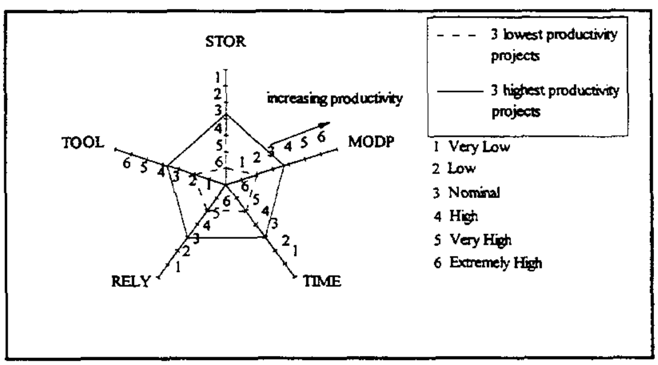


Figure 6. Scores of productivity factors for highest and lowest productivity companies in ESA dataset

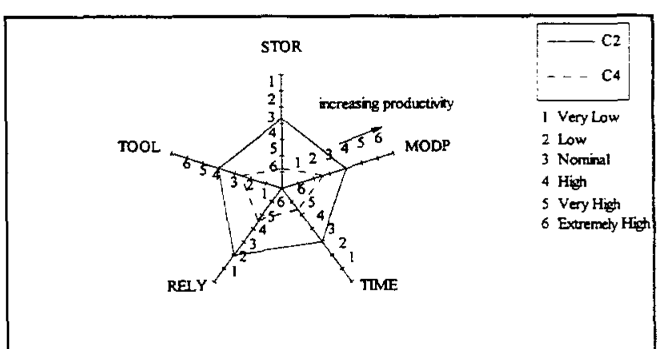


Figure 7. **Scores of productivity factors by highest and lowest productivity categories in ESA dataset**

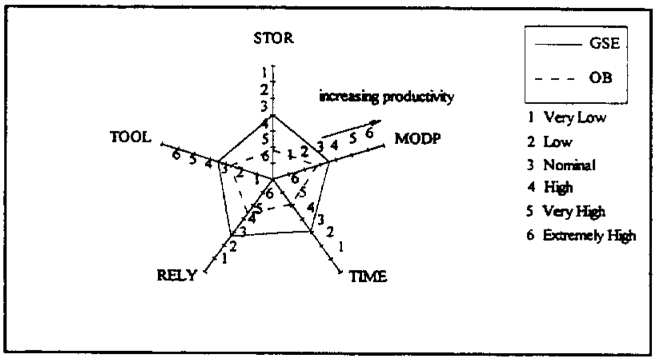
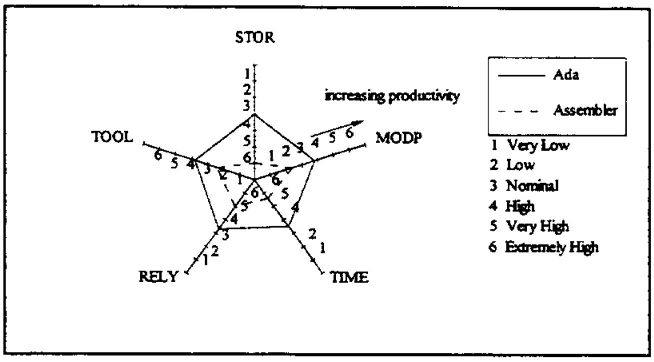


Figure 8. **Scores of productivity factors by highest and lowest productivity languages in ESA dataset**



Phase II - ESA Dataset Productivity Models based on Combinations of Continuous and Class Variables

In the second phase of the analysis, the best productivity models based on combinations of continuous and class variables were identified. The variables MODP and TOOL were found to be significantly correlated at the 0.81 level and thus were not considered together as independent variables in the multivariate regression.

Multi-variable models were run to determine what combinations of class variables, the system size (KLOC), and the seven COCOMO factors could explain the most variation in productivity. Improvement in some of the models was obtained by including the team size. However, as the purpose of this productivity analysis was to identify factors that can be used to predict effort, these productivity models are not presented[2]. A summary of the results is shown in Table 4 where NOBS refers to the number of observations, ROOT MSE is the root mean squared error of the model, and MODEL(significance) describes the type of model,

[2]Effort being by definition a function of team size and duration.

additive or log, and its significance. The 2-class productivity models based on Country do not make much sense because the productivity values for each country are based on too few observations. The best 2-class model is based on Language, Category and reliability requirements. This model explains 90% of the productivity variance and is significant at the .0001 level. As the objective of this analysis was to develop simple effort estimation models based on productivity factors which can be easily applied to any company, the best productivity models based on 1 class and no class variables are also shown. It can be seen that the one-class models, which explain 89% of the productivity variance and are significant at the .0001 level, are nearly as good as the two-class model.

Table 4. Summary of productivity analysis of best mixed models for ESA dataset

VARIABLES	NOBS	VARIANCE EXPLAINED	ROOT MSE	MODEL (significance)
subset Category* subset Country, RELY, TOOL	54	96%	0.26	Log(.0001)
subset Category* subset Country RELY, MODP	53	95%	0.29	Log(.0001)
subset Language* subset Category RELY	**60**	**90%**	**0.39**	**Log(.0001)**
subset Category, STOR, RELY, TOOL	55	89%	0.35	Log(.0001)
subset Category STOR, RELY, MODP	55	89%	0.35	Log(.0001)
STOR, RELY, TOOL	59	70%	0.52	Log(.0001)
STOR, RELY, MODP	59	70%	0.53	Log(.0001)

* *Signifies crossed effect of class variables*

Phase III - Results of Development of Effort Estimation Model based on Significant Productivity Factors for ESA Dataset

In the third phase of the analysis, the best effort estimation models based on significant productivity factors were identified. The best effort prediction model was found to be based on the size of the project with the addition of the main factors found to effect productivity: application category, language and software reliability requirements (see Table 5). This model has an R-squared of 0.95 and is significant at the .0001 level. The average estimation error was 20% and 67% of the estimations were within 25% of the actuals. The correlation coefficient of 0.95 shows that the model is very consistent. Effort estimation models based on the duration of the project, the start year of past projects, or the maximum team size were rejected as not being useful to predict the effort of future projects. The five best ESA models presented are of three types: 2-class, 1-class and no-class in order to enable each company to determine which general ESA model best fits their data. It can be seen that the 1-class models, with an R-squared of 0.93 and a significance level of .0001, are nearly as accurate as the 2-class models. The effort estimation equations for each model can be found in Appendix 1. A model based on KLOC only is shown for comparison.

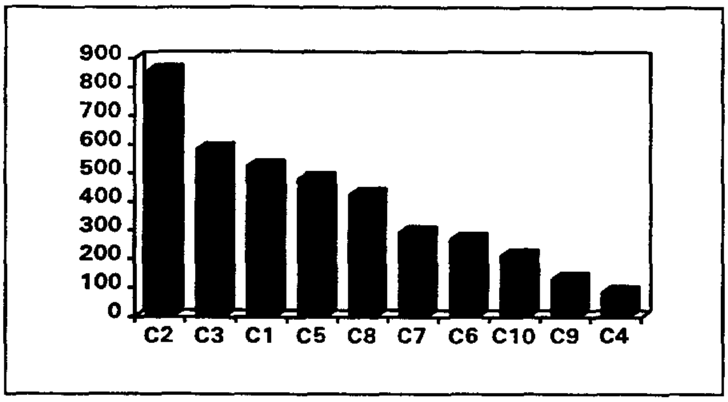
Table 5. ESA effort prediction models

VARIABLES	NOBS	R**2	ROOT MSE	MODEL (sign.)	MMRE	CORR	PRED(.25)
subset Category* subset Language, KLOC, RELY	**60**	**0.95**	**0.39**	**Log(.0001)**	**20%**	**0.95**	**67%**
subset Category, KLOC, STOR, RELY, MODP	55	0.93	0.35	Log(.0001)	26%	0.94	62%
subset Category, KLOC, STOR, RELY, TOOL	55	0.93	0.35	Log(.0001)	26%	0.93	60%
KLOC, STOR, RELY, MODP	59	0.82	0.52	Log(.0001)	40%	0.95	51%
KLOC, STOR, RELY, TOOL	59	0.82	0.51	Log(.0001)	41%	0.94	46%
KLOC	77	0.50	0.85	Log(.0001)	79%	0.74	22%

Phase IV - Company Specific Productivity Analysis Results of Individual Class Variables

The fourth phase of the analysis was concerned with determining which individual variables explain the greatest amount of productivity variation of company C10. Figure 9 shows the mean productivity of company C10 in relation to other companies in the ESA database which supplied data for three or more projects.

Figure 9. Mean productivity by company (LOC/MM)



The percentage of productivity variance explained by individual class variables for company C10, compared with the ESA dataset, is shown in Figure 10. Limiting Language to the subset of 3 languages used in 3 or more projects caused the amount of variance explained by language for company C10 to decrease from 68% to 63%. The mean productivity of each language for company C10 and the ESA dataset are presented in Figures 11 and 12. Ada is shown to be a high productivity language and Coral (COR) and Assembler (AS) low productivity languages for both datasets. No variation was explained by the different categories of projects in company C10, even when the subset of 3 categories with 4 or more observations was considered. This may be because most of the projects undertaken by company C10, and the entire category subset, were in the medium productivity categories of GRD, RT and SIM (Maxwell et al 1994). The mean productivity of each category for company C10 and the ESA dataset are shown in Figures 13 and 14. The Ground Control category has a lower productivity for company C10 than for the ESA dataset.

Figure 10. Percentage of productivity variance explained by individual class variables

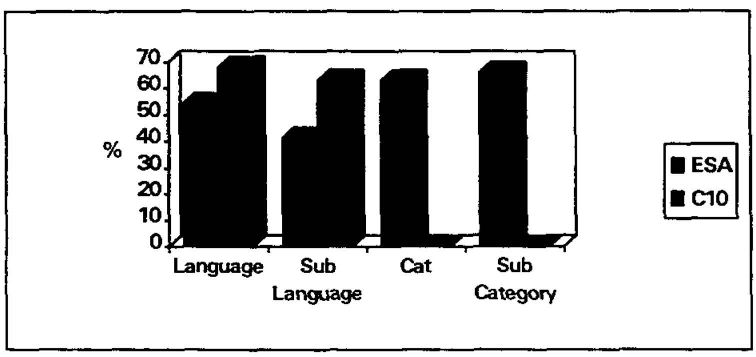
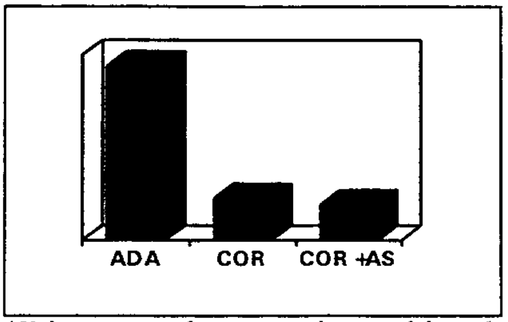
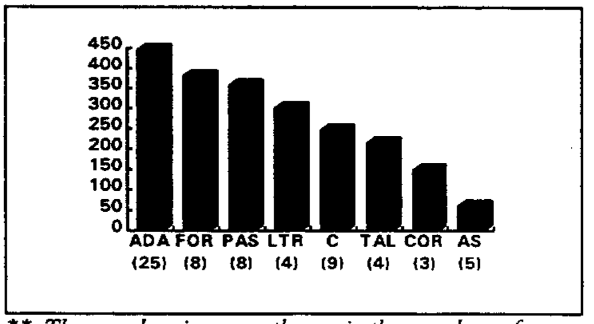


Figure 11. **Mean productivity by language* Company C10**



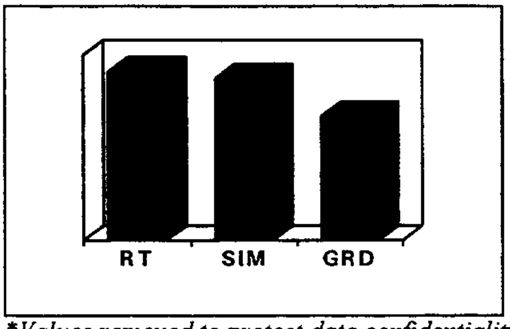
**Values removed to protect data confidentiality*

Figure 12. **Mean productivity by language (LOC/MM)** ESA dataset**



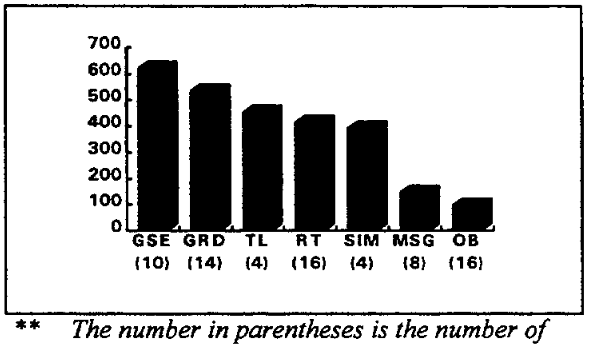
*** The number in parentheses is the number of observations.*

Figure 13. **Mean productivity by category* Company C10**



**Values removed to protect data confidentiality*

Figure 14. **Mean productivity by category (LOC/MM)** ESA dataset**



*** The number in parentheses is the number of observations.*

Company Specific Productivity Analysis Results of Individual Continuous Variables

The percentage of productivity variance explained by individual continuous variables for company C10 and the ESA dataset are presented in Figures 15 and 16. It can be seen that the factors which explain the productivity variation of company C10 differ considerably from the ESA dataset. Productivity was found to significantly decrease with increasing project duration and team size, and to significantly increase over time for company C10. Company C10 productivity was also found to significantly increase with increasing use of modern programming practices and tools, and with decreasing storage constraints and virtual machine volatility. The non-significance of required software reliability is due to its not varying much in the dataset.

Figure 15. Percentage of productivity variance explained by individual continuous variables

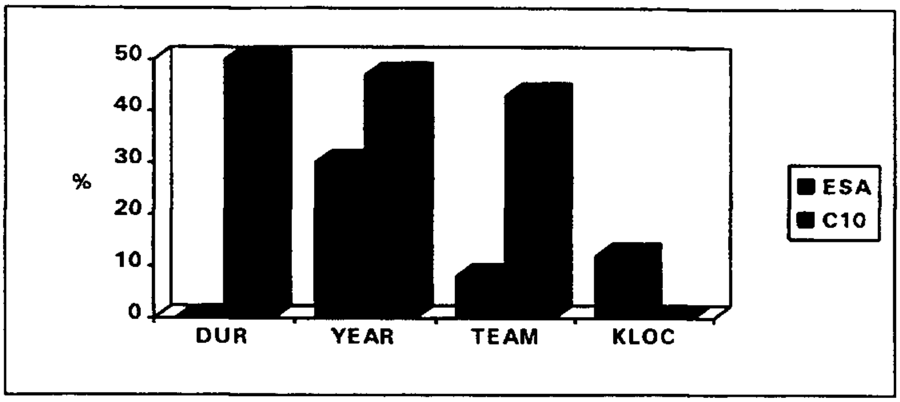


Figure 16. Percentage of productivity variance explained by 7 COCOMO factors

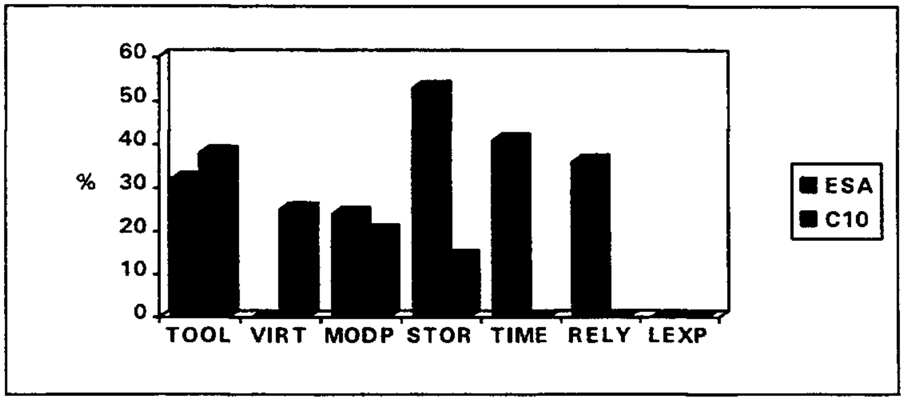


Figure 17 shows the average score of the significant productivity factors MODP, TOOL, VIRT and STOR for the 3 highest and 3 lowest productivity software projects. The average scores of these productivity factors for the subset of 3 languages for which there were a sufficient number of observations is shown in Figure 18. Further evidence of the usefulness of plotting the scores of the four significant COCOMO factors is shown in Figure 19. Breakdown by project category was not found to significantly affect the productivity of company C10; this is also shown by the similarity of the average productivity factor scores for the three categories. Significant relationships were also found between MODP, TOOL, VIRT and STOR and time. Modern programming practices and tool use have been increasing over time, while storage constraints and virtual machine volatility have been decreasing over time for company C10.

Figure 17. Scores of productivity factors for highest and lowest productivity projects of Company C10

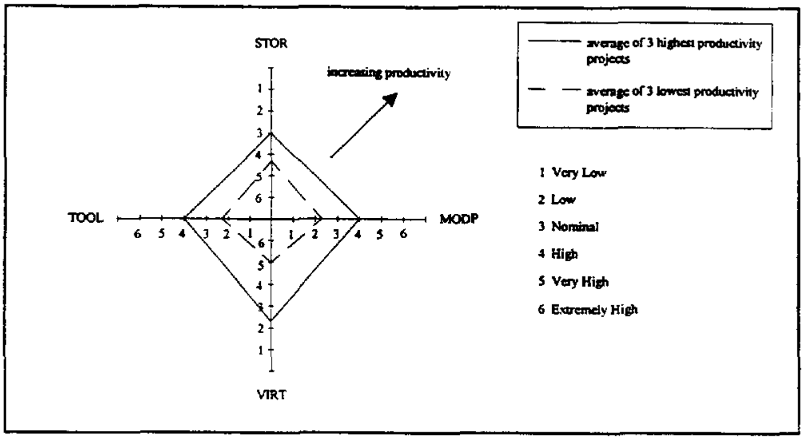


Figure 18. Scores of productivity factors by language - Company C10

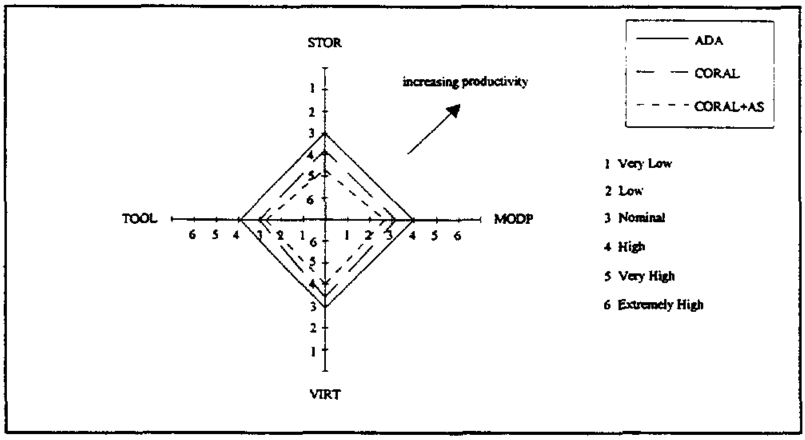
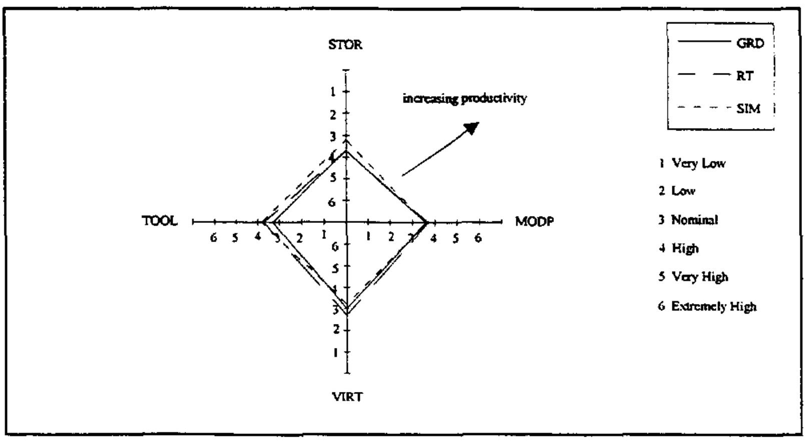


Figure 19. Scores of productivity factors by category - Company C10



Phase V - Company Specific Productivity Models based on Combinations of Continuous and Class Variables

In the fifth phase of the analysis, the best company specific productivity models based on combinations of continuous and class variables were identified. Analysis of Variance Models based on combinations of class variables were not considered as categorising the data by two class variables led to too many cells with only one observation. Multi-variable models were run to determine what combinations of the system size (KLOC), the seven COCOMO factors, and the class variable language could explain the most variation in productivity of company C10. Improvement in some of the models was obtained by including duration. However, as the purpose of this productivity analysis was to identify factors that can be used to predict effort, these productivity models are not presented. The best company specific model was based on language and virtual machine volatility. This model explained 80% of the variance and was significant at the .0001 level. The best productivity model for company C10 is given in Table 6.

Table 6. Best productivity model for company C10

VARIABLES	NOBS	VARIANCE EXPLAINED	ROOT MSE	MODEL (significance)
subset Language, VIRT	19	80%	0.47	Log (.0001)

Phase VI -Results of development of Effort Estimation Model based on significant productivity factors for company C10

The best company C10 effort estimation models based on significant productivity factors were identified and compared with general ESA models in the sixth phase of the analysis. The best company specific effort prediction model was found to be based on the size of the project with the addition of the main factors found to effect productivity: development language and virtual machine volatility (see Table 7). This model had an R-squared of 0.94 and was significant at the .0001 level. The average estimation error was 29% and 63% of the estimations were within 25% of the actuals. The correlation coefficient of 0.98 shows that the model is very consistent. Effort estimation models based on the duration of the project, the start year of past projects, or the maximum team size were rejected as not being useful to predict the effort of future projects. A company C10 model based on KLOC alone is shown for comparison.

Table 7. Best effort prediction models developed for company C10

VARIABLES	NOBS	R**2	ROOT MSE	MODEL (sign.)	MMRE	CORR	PRED(.25)
KLOC, subset Language, VIRT	**19**	**0.94**	**0.48**	**Log (.0001)**	**29%**	**0.98**	**63%**
KLOC	29	0.70	0.88	Log(.0001)	84%	0.87	17%

Other Effort Estimation Models applied to Company C10

General effort estimation models developed using the ESA dataset were used to predict effort for company C10. Although none of these models were extremely accurate, the best general effort prediction model was the ESA no-class model based on KLOC and 3 productivity factors: RELY, STOR and TOOL. The average estimation error was 50% and 23% of the estimations were within 25% of the actuals. The correlation coefficient of 0.92 shows that the model is very consistent. The ESA 2-class model based on KLOC, language, category and 1 productivity factor: RELY gives comparable results; however, only 12 projects used a language and category for which there was a corresponding ESA equation. These models are presented in Table 8.

Table 8. Results of predicting company C10 effort with general effort estimation models

MODEL	VARIABLES	NOBS	MMRE	CORR	PRED(.25)
ESA no class model	**KLOC, RELY, STOR, TOOL**	**26**	**50%**	**0.92**	**23%**
ESA 2-class model	subset Category* subset Language, KLOC, RELY	12	46%	0.83	25%
ESA KLOC model	KLOC	29	60%	0.84	17%
ESA 1-class model	subset Category, KLOC, STOR, RELY, TOOL	26	58%	0.87	12%

Interpretation of Results: Management Implications

The effective management of software development continues to be an important challenge for managers. There are two major factors contributing to this challenge. Firstly, the complexity and criticality of software within industry is high and continues to grow significantly every year as software becomes an increasingly important component in many products. The amount of software code in most consumer products is currently estimated to be doubling every two years (Gibbs 1994). Secondly, despite significant progress in computing over the last decades, software development continues to be perceived as more of an art than a science in most organizations.

The immaturity of software development is reflected in the abundance of stories about major software disasters. Recent research by Lederer and Prasad (1992) found that two thirds of all major software projects substantially overran their estimates. Surveys undertaken by the Software Engineering Institute at Carnegie Mellon University show that the vast majority of organizations (over 75%) are stuck at level 1 of their Capability Maturity Model (Gibbs 1994). At level 1 of this five level software process maturity model, the organization is characterised as having an ad hoc, or possibly chaotic, process (Marciniak 1994). It is not uncommon for managers to use guessing and intuition to estimate different aspects of a software development project. However, this practice has been found to be highly positively correlated with the percentage of large projects overrunning their estimates (Lederer and Prasad 1992).

In this paper we have addressed three important issues facing managers in charge of estimating the effort of software development projects:

1. We have determined that simple effort estimation models based on KLOC and the significant productivity factors of a given database are reasonably accurate.
2. We have confirmed the hypothesis that accurate effort estimation is possible with a very small number of productivity factors.
3. We have shown that effort predictions made for an individual company's software projects using a general model based on a multi-company database were less accurate than predictions made using a company specific model.

The management implications of these findings are that it is best for organizations to build up their own software development database based on the factors which they believe are responsible for the productivity variation of their projects. When this database contains about 20 projects, it should be analysed to determine which factors are really causing the productivity variation. An effort estimation model can then be developed using these significant productivity factors. Further data collection can then be limited to a smaller number of productivity factors, keeping in mind that the significance of productivity factors may vary over time in a particular company. If the creation of such a database is not possible, the application of general models may be more accurate than the use of guessing and intuition.

Conclusions

The objectives of this paper were firstly, to develop and evaluate simple general effort estimation models for European space, military and industrial software applications based only on those factors found to significantly impact the productivity of these types of projects; and secondly, to determine if models based on a multi-company database can be successfully used to make effort estimations within a specific company. The later was accomplished by developing a simple company specific effort estimation model based only on those factors, common to the ESA dataset, which had a significant impact on this company's productivity; and by comparing the accuracy and consistency of the effort estimates of the company specific model to the general ESA models. The development and testing of the effort estimation models is illustrated in Figure 1. To our knowledge, no other published research has yet developed and analysed software development effort estimation models in this way.

Our study confirms the hypothesis that only a small number of productivity factors are necessary to develop an accurate effort estimation model within one environment. We found that the best general effort estimation models were based on the size of the project and the main factors found to effect the productivity of the ESA dataset: application category, language, required software reliability, main storage constraint, and the use of modern programming practices or software tools. The best company specific effort estimation model was based on the two major factors, common to the ESA dataset, which explained the productivity of the individual company: language and virtual machine volatility.

Also of managerial interest is the fact that we have determined that effort predictions made for the individual company's projects using the general models were a great deal less accurate than the company specific model; on average a 50% estimation error as opposed to 29% error for the company specific model. However, it is likely that in the absence of enough resources and data for a company to develop its own model, the application of general models may be more accurate than the use of guessing and intuition. We estimate that around 20 projects would be needed to develop a company specific effort estimation model.

It should also be stressed that the company specific model developed in this study was based only on significant productivity factors common to the ESA dataset. Accuracy of this model could certainly be further improved through the development of effort estimation models taking into account all of the factors which may effect the productivity in the company. Further research should concentrate on studying the reasons for productivity variations in individual companies.

References

Albrecht, A.J. and J.E. Gaffney, Jr., "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation", *IEEE Transactions on Software Engineering*, Vol. SE-9, No. 6, (Nov. 1983), 639-648.

Aron, J.D., "Estimating Resources for Large Programming Systems", *Software Engineering: Concepts and Techniques,* Edited by J.M. Buxton, P. Naur and B. Randell, Litton Education Publishing, Inc., (1976), 206-217.

Bailey, J.W. and V.R. Basili, "A Meta-Model for Software Development Resource Expenditures", *Proceedings of the Fifth International Conference on Software Engineering*, San Diego, CA, (1981), 50-60.

Basili, V.R., "Resource Models", *Tutorial on Models and Metrics for Software Management and Engineering,* IEEE Catalog No. EHO167-7, (1980), 4-9.

Basili, V.R. and K. Freburger, "Programming Measurement and Estimation in the Software Engineering Laboratory", *The Journal of Systems and Software 2*, (1981), 47-57.

Banker, R.D., S.M. Datar and C.F. Kemerer, "A Model to Evaluate Variables Impacting the Productivity of Software Maintenance Projects", *Management Science*, Vol. 37, No. 1, (January 1991), 1-18.

Boehm, B.W., *Software Engineering Economics*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1981.

Boehm, B.W., "Improving Software Productivity", *IEEE Computer,* vol. 20, (September 1987), 43-57.

Conte, S.D., H.E. Dunsmore and V.Y. Shen, *Software Engineering Metrics and Models*, The Benjamin/Cummings Publishing Company, Inc., Menlo Park, California, 1986.

Cusumano, M.A. and C.F. Kemerer, "A Quantitative Analysis of U.S. and Japanese Practice and Performance in Software Development", *Management Science*, Vol. 36, No. 11, (November 1990), 1384-1406.

DeMarco, T., *Controlling Software Projects*, Yourdon Press, New York, N.Y., 1982.

Gibbs, W.W., "Software's Chronic Crisis", *Scientific American*, (September 1994), 72-81.

Heemstra, F.J., "Software Cost Estimation", *Information and Software Technology,* Vol. 34, No. 10, (Oct. 1992), 627-633.

Kemerer, C.F., "An Empirical Validation of Software Cost Estimation Models", *Communications of the ACM,* Vol. 30, No. 5, (May 1987), 416-429.

Kitchenham, B.A., "Empirical studies of assumptions that underlie software cost-estimation models", *Information and Software Technology*, Vol. 34, No. 4, (April 1992), 211-218.

Lederer, A.L. and J. Prasad, "Nine Management Guidelines for Better Cost Estimating", *Communications of the ACM,* Vol. 33, No. 2, (Feb. 1992), 51-59.

Matson, J.E., B.E. Barrett and J.M. Mellichamp, "Software Development Cost Estimation Using Function Points", *IEEE Transactions on Software Engineering,* Vol. 20, No. 4, (April 1994), 275-287.

Marciniak, J.J., *Encyclopedia of Software Engineering, Vol. 2*, John Wiley & Sons, Inc., New York, 1994.

Maxwell, K., L. Van Wassenhove and S. Dutta, "Software Development Productivity of European Space, Military and Industrial Applications", *INSEAD Working Paper*, (1994).

Mohanty, S.N., "Software Cost Estimation: Present and Future", *Software-Practice and Experience*, Vol. 11, (1981), 103-121.

Mukhopadhyay, T. and S. Kekre, "Software Effort Models for Early Estimation of Process Control Applications", *IEEE Transactions on Software Engineering*, Vol. 18, No. 10, (Oct. 1992), 915-924.

Mukhopadhyay, T., S.S. Vicinanza and M.J. Prietula, "Examining the Feasibility of a Case-Based Reasoning Model for Software Effort Estimation", *MIS Quarterly*, (June 1992), 155-171.

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, (July 1978), 345-361.

Vosburgh, J, B. Curtis, R. Wolverton, B. Albert, H. Malec, S. Hoben and Y. Liu, "Productivity Factors and Programming Environments", *Proceedings of the Seventh International Conference on Software Engineering*, (1984), 143-152.

Walston, C.E. and C.P. Felix, "A method of programming measurement and estimation", *IBM Systems Journal* 16, No. 1, (1977), 54-73.

Acknowledgements

The authors would like to thank the European Space Agency for their funding of this project and Dr. Benjamin Schreiber in particular for initiating the data collection effort. We would also like to thank the data provider from company C10 for his data and comments.

TABLE 1: Overview of some productivity factors considered in past research.

Some Major Productivity Factors	ESA Data	A*	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
Country	X											X							
Company	X														X				
Category/Type	X											X	X		X		X		
Industrial/Business Environment	X												X		X				
Language	X	X	X			X							X	X					
Team Size	X						X			X	X								
Duration	X	X					X									X	X		
Project Size	X		X			X			X		X		X				X	X	
Required Software Reliability	X							X											
Execution Time Constraint	X							X	X									X	X
Main Storage Constraint	X							X	X									X	X
Virtual Machine Volatility	X							X											
Programming Language Experience	X			X				X	X					X					X
Modern Programming Practices	X		X	X	X			X	X	X			X	X	X	X		X	X
Tool Use	X				X			X		X			X	X		X			
Product Complexity				X				X	X					X	X	X		X	X
Analyst Capability					X			X						X		X			
Applications Experience				X	X			X	X					X		X			X
Programmer Capability				X	X			X						X	X				
Virtual Machine Experience				X		X		X	X					X					X
Amount of Documentation							X			X									X
Overall Personnel Experience										X			X	X				X	X
Customer Interface Complexity				X					X					X				X	X
Design Volatility				X					X	X				X		X		X	X
Hardware Concurrent Development									X									X	X
Quality Assurance					X					X						X			
Development Environment (On-line)			X			X								X	X				

* Table 2 describes the research referred to by each letter.

TABLE 2: Overview of major databases which include productivity factors.

Database Code in Table 1	Reference	No. Projects	Environment	Geographical Scope	Productivity Measure
ESA Data	Our research	99	Space/Military/Industrial	Europe	L.O.C.
A	Aaron 1976	9	IBM Large Systems	USA	L.O.C.
B	Albrecht 1979	22	IBM Data Processing	USA	F.P.
C	Bailey and Basili 1981	18	Space (NASA/Goddard)	USA	L.O.C.
D	Banker et al. 1991	65	Commercial Maintenance Projects	USA	F.P. and L.O.C.
E	Behrens 1983	24	Data Processing	USA	F.P.
F	Belady and Lehman 1979	37	Not Identified	USA	L.O.C.
G	Boehm 1981	63	Mixture	USA	L.O.C.
H	Brooks 1981	51	Mixture (from Walston-Felix)	USA	L.O.C.
I	Card et al. 1987	22	Space (NASA/Goddard)	USA	L.O.C.
J	Conte et al. 1986	187	Mixture	USA	L.O.C.
K	Cusumano and Kemerer 1990	40	Mixture	USA and Japan	L.O.C.
L	Jones 1991	4000	Mixture (primarily Systems and MIS)	USA	F.P. (converted from L.O.C.)
M	Kitchenham 1992	108	Not Identified (probably Commercial)	Europe (1 company)	F.P.
N	Lawrence 1981	278	Commercial	Australia	L.O.C.
O	Nevalainen and Mäki 1994	120	Commercial	Finland	F.P.
P	Putnam and Myers 1992	1486	Mixture (primarily Business Systems)	Primarly USA, also Canada, Western Europe, Japan and Australia	L.O.C.
Q	Vosburgh et al 1984	44	Mixture (from ITT)	9 countries	L.O.C.
R	Walston and Felix 1977	60	Mixture (from IBM)	USA	L.O.C.

Appendix 1
General ESA Effort Estimation Equations

ESA 2-class model

$$Effort = 1.6 \times LC \times KLOC^{0.92} \times RELY^{1.15}$$

where LC is the Language/Category Multiplier given in Table 8. In the event that there is no value of LC given for the needed Language and Category the 1-class or no-class ESA models should be used.

Table 8. Value of multiplier LC for each language and category

Lang / Category	GSE	GRD	TL	RT	SIM	MSG	OB
Ada	0.49	0.28	0.52	0.49	0.49	0.79	1.10
Fortran	0.57	0.35		0.63	0.47		
Pascal	0.44	0.17	0.81	0.41		0.69	1.86
LTR		0.38		0.84			1.23
C		0.54	1.40		1.02		1.16
TAL		0.15				1	
Coral				0.73		1.22	
Assembler							2.04

ESA 1-class models

$$Effort = 1.5 \times CAT1 \times KLOC^{1.02} \times STOR^{0.73} \times RELY^{0.87} \times MODP^{-0.73}$$

$$Effort = 2.1 \times CAT2 \times KLOC^{1.00} \times STOR^{0.67} \times RELY^{0.84} \times TOOL^{-0.88}$$

where CAT1 and CAT2 are the Category Multipliers defined in Table 9.

Table 9. Value of CAT1 and CAT2 for each Category.

Category	CAT1	CAT2
GRD	0.35	0.38
GSE	0.59	0.57
MSG	1.33	1.46
OB	1.32	1.27
RT	0.67	0.66
SIM	0.71	0.71
TOOL	1.00	1.00

ESA no-class models

$$Effort = 1.4 \times KLOC^{0.88} \times RELY^{0.83} \times STOR^{1.23} \times MODP^{-0.93}$$

$$Effort = 2.0 \times KLOC^{0.88} \times RELY^{0.89} \times STOR^{1.06} \times TOOL^{-1.11}$$