Analysis of Factors that Affect Productivity of Enterprise Software Projects

Tsuneo Furuyama
Scool of Science, Tokai University
Kitakaname 4-1-1, Hiratsuka city, Japan
furuyama@tokai-u.jp

Abstract

Analysis of the factors affecting the productivity of enterprise software projects the data of which had many missing values revealed five important findings. (1) Productivity was proportional to the root of the fifth power of test case density and the root of the fifth power of fault density. (2) Developing software with a high level of security or reliability are the main causes of low productivity. (3) Developing software with a document-generation tool or under the condition that the objectives and priorities are very clear, are the main causes of high productivity. (4) The productivity of projects managed by skilled project managers tends to be low because they tend to develop software with large-scale and rigorous requirements and try to detect as many faults as possible. (5) It is important to provide an environment with sufficient working space or to use a development framework when developing software that requires a high level of security.

1. Introduction

The baselines of software productivity differ with the type of software. For example, the productivity of basic software, such as operating systems, is lower than that of application software because many more verifications and tests are needed during development to ensure high reliability. A much larger development staff is needed when developing extremely large software systems than when developing mid-size software systems, resulting in dramatically larger overhead and lower productivity.

Software productivity across many projects varies widely even if the same type of software is developed because many factors related to software projects affect their productivity. It would be helpful to know the factors that affect productivity when estimating the productivity of development efforts in the project planning stage, even though they are not given systematically as models or formulations. It would also be helpful if those factors can be controlled by the project manager (PM) when establishing and managing a project. Moreover, if the cause-effect relationships between the factors can be clarified and quantified, quantitative models or formulations could be established.

Generally, cost (or effort) estimation models, so-called cost models, have been proposed and discussed instead of those for estimating productivity because productivity is defined as the ratio of software size generally given beforehand and the total amount of effort. Many cost models with various estimation methods have been proposed, such as multiple regression analysis, analogy, expert judgment, summing up each cost of work breakdown structure (WBS), and the approach of artificial intelligence technologies such as neural networks or Bayesian networks.

Among various types of models, models applying multiple regression analysis with an experience database have been extensively studied and have been the majority even in recent years [1]. Dejaeger et al. compared these various types of models and concluded that ordinary least squares regression in combination with a logarithmic transformation performed the best [2].

The most well-known models of this type are the Cost Construction Model (COCOMO) [3] and its successive model, COCOMO II [4], with which the estimation formulae were established. In COCOMO, software development efforts can be estimated using the formula with which software development efforts are approximated by software size raised to the power of a factor and multiplied by 15 coefficients of cost drivers. The power value of the factor ranges from 1.05 to 1.2 in accordance with the development circumstances. In COCOMO II, five factors that affect the power of size were defined as so-called scale factors, different from the original cost drivers defined in COCOMO. Other improvements with COCOMO II were inclusion of the function point (FP) as a new size measure in addition to source lines of code (SLOC) and reused code size into the model.

Other than COCOMO, many cost models have been proposed and discussed [5], [6], [7], [8]. Maxwell et al. presented an overview of some productivity factors considered in past researches including their own research and COCOMO, and also an overview of major databases that include productivity as was the case in 1996 [5]. The largest set of productivity factors among the 19 studies presented in their paper consisted of 15 elements. Maxwell also analyzed the database including 206 projects in Finland that had 15 ordinal factors different from the COCOMO factors and identified significant variables affecting productivity for 5 business sectors [6]. Sentas et al. constructed three cost models with 4 to 6 ordinal variables based on three databases: the Maxwell, COCOMO81, and ISBSG7, respectively [7].

Reliable and widely acceptable factors affecting productivity can be identified only from applying appropriate analysis methods to abundant and reliable data collected by reliable organizations. This means that an experience database or dataset has an important role constructing cost models regardless of the analysis methods and theories for analysis. Providing appropriate variables and collecting a large amount of reliable data are possible to reveal new effective factors and improve the precision of the coefficients of these factors.

In the Predictor models in software engineering (PROMISE) repository [9], 17 databases are open to the world. The majority is related to COCOMO systems, such as the coc81 and coc2000 databases. Table 1 gives an overview of the subset of the PROMISE repository selected from the viewpoint of its size; more than or equal to 50 projects. Variables ("attributes" in the repository) of the nasa93 database are the same as those of the coc81 database, which has 17 variables including project identifiers (ID), 15 of which are subject to ordinal scales. For example, the variable: required software reliability (rely) in the nasa93 database has six levels: very low, low, neutral, high, very high, and extremely high. Variables of the nasa93-dem database are the same as those of the coc81-dem database, which has 25 variables, 17 of which are subject to ordinal scales. The Maxwell database has 26 variables, 15 of which are subject to ordinal scales. The cosmic database, which is not included in Table 1 since its size is 42, is a subset of the ISBSG database. It has at least 86 variables, but most are details of efforts, defects, and project profiles subject to nominal scales, and the variables subject to ordinal scales seem to be less than 10 even if variables such as the CASE tool, whose choice is "yes" or "no", are included in this category. The China and Kitchenham databases have data collected from 499 and 145 projects, respectively, but most of their variables are those related to FP counting, and no variable related to ordinal scales can be found. The usp05 database has the data collected from 919 projects, but these projects are not of industries but university student projects, and 83 missing values are included in the database.

Of the nine databases listed in Table 1, seven are complete, in other words, they have no missing values, and only two are almost complete. Most cost models derived from regression analysis are based on the complete databases because of the difficulty in constructing cost models by using incomplete databases. For the dataset with missing values, list-wise deletion is generally used to complete the dataset. However, this method sometimes reduces the variables available too much to construct reliable cost models for small databases.

The Software Reliability Enhancement Center (SEC) of the Information-Technology Promotion Agency (IPA) in Japan has been collecting enterprise software project data (SEC database) that include qualitative variables, such as the skill level of the project staff, as well as quantitative variables such as software size (size), development effort (effort), and development duration (duration). The strengths of the SEC database are that more than 200 variables are defined even excluding detailed variables, and the size of this database is more than 3000. The weakness includes a large amount of missing values, as discussed later.

By effectively using the SEC database, a precise model can be constructed. The multiple regression analysis after list wise deletion is the simplest approach to analyze data. However, for the SEC database, this approach seems difficult because of the following reasons. If high priority is given to the number of projects when making a complete subset, some important variables may be lost before analysis, or if high priority is given to the number of variables, the number of selected projects is not sufficient for analysis. Another possible method with multiple regression analysis is a step-by-step progressive selection of variables without prior selection of complete subset data. However, this method also seems difficult from my experience because the subset data in every step are different so that the combinations of variables unselected in the prior steps may be more appropriate than the current combinations. Statistical tools are not recommended for the dataset with missing values.

Some statistical methods, such as the multiple imputation method for interpolating missing values to complete the dataset, have been proposed to solve this problem. For missing data in the category of missing completely at random (MCAR) or missing at random (MAR), these methods are effective. However, for missing values in the category of missing not at random (MNAR) or too many missing values, interpolation is not effective.

The SEC database is growing yearly. Useful applications to this database to construct precise cost models are now expected. This paper presents the results of the first step of the analysis for the SEC database by applying multiple regression analysis and analysis of variance to the SEC database. For quantitative variables, the number of test cases and number of faults in addition to size in FP were selected to analyze the effects on efforts in the test phases. For

Table 1. Overview of PROMISE database

Name of DB		coc81	coc81-dem	coc2000	nasa93	nasa93-dem	Maxwell	usp05	China	Kitchen-ham
Number of projects		63	63	125	101	93	62	919	499	145
Attributes		19	27	50	24	27	27	17	19	10
	Assigned to value in ordinal scale	*15	22	*22	15	22	15	2	0	0
Number of missing values		0	0	0	0	0	0	83	0	13

Note: Databases including more than or equal to 50 projects are listed.
* Value of each variable is selected among a few pre-defined numeric constants.

qualitative variables, 39 variables were selected to analyze the effect of each variable alone after pair wise-deletion. Also, cross effects of these variables were attempted.

Section 2 describes the analyzed data, and Section 3 describes the statistical analysis methods. In Section 4, the analysis results, that is, the factors affecting productivity when developing enterprise software, are described. In Section 5, the factors are studied to determine whether they actually affect productivity by comparing the present findings to previous findings, well-known axioms, and experience. The meanings of the factors are also discussed. Section 6 summarizes the main results.

2. Analysis data

2.1. Overview

Data to be analyzed were collected from 3089 enterprise software development projects by the IPA SEC from fiscal year 2004 to 2012 [10]. The data for the 1213 projects developing new software were analyzed first because the variations were smaller than those for projects modifying software, meaning that the analysis results were more precise. The data for 523 of the 1213 projects were used from the conditions of performing five phases (fundamental design or FD, detail design, manufacture, system test, and total test by vendor) and including the following variables: software size based on FP and development effort. The definitions of the variables are given in "White Paper 2012-2013 on Software Development Projects in Japan" [10].

2.2. Criterion and predictor variables

(1) Quantitative predictor variables

The criterion variable is effort and the predictor variables are size based on FP, number of test cases, which is defined as the sum of the number of test cases in the system test and in the total test phase done by the vendor, and number of faults detected during development (faults), which is defined as the sum of the number of faults detected in the system test phase and number detected by the vendor in the total test phase. Table 2 lists the fundamental log values for the quantitative variables. Table 3 is a matrix of correlation coefficients for four quantitative variables. The values for the quantitative variables were statistically analyzed after common logarithmic transformation (logarithmic transformation) for the reason explained in Section 3.1. Hereafter, all quantitative variables and their analysis results are those after logarithmic transformation except in the case of special descriptions.

Table 2. Fundamental log values of quantitative variables

	Effort /FP	Effort	FP	Number of test cases	Number of faults
Mean	0.988	3.858	2.870	3.133	2.061
Standard deviation	0.368	0.667	0.489	0.754	0.663
Number of projects	523	523	523	324	310

Note: Unit of effort is person-hours.

Table 3. Matrix of correlation coefficients for four quantitative variables

	Effort	FP	Number of test cases	Number of faults
Effort	-	0.841	0.570	0.674
FP	523	-	0.653	0.706
Number of test cases	324	324	-	0.701
Number of faults	310	310	288	-

Note: Upper right shows correlation coefficients.
Lower left shows number of projects.

(2) Qualitative predictor variables

The criterion variable is productivity, which is defined as the ratio of effort and size before logarithmic transformation. Note that the smaller the ratio, the higher the productivity. The variables identified as candidate predictors satisfy two conditions: 1) the number of projects for which data were collected is equal to or more than 50 and 2) all categories or levels have more than or equal to 10 data values and 15% of the total amount of data values of the variable.

Table 4 shows the 39 candidate qualitative predictor variables. The industry area, computer architecture, and programming languages mainly used were excluded because the median values (which are equivalent to the mean values of the data after logarithmic transformation) of their categories are already available in "White Paper 2012-2013 on Software Development Projects in Japan" [10], and the order of each category in terms of productivity has been clarified.

Table 5 shows missing value ratios for each qualitative variable. All ratios of missing values for each variable were more than half. The maximum ratio of missing values was 0.84 for user's business experience, minimum was 0.53 for evaluation of cost plan, and mean ratio was 0.69.

Table 6 shows the number of values each project has among 38 qualitative variables (note that two variables related to the test team are derived from the same original data as described later). While 38 projects had all values for the qualitative variables selected for this analysis, 114 projects had no values. Since these 114 projects were omitted when analyzing qualitative data, 409 projects were used for qualitative analysis.

Table 4. Candidate qualitative predictor variables

Category	Variables
Overall project	Introduction of new technology/Clarity of role assignments and each staff member's responsibilities/Clarity of objectives and priorities/Quantitative quality criteria for delivery/Quality assurance system in fundamental design (FD) phase
	Working space/Noise conditions
Evaluation of plan	Evaluation of plan (cost/duration/ quality)
Tool usage	Similar project/Project management tool/Configuration control tool/Design support tool/Document-generation tool/ Debug and test tool/Code generator/ Development framework
User's skill levels and commitment*	User's commitment to defining requirement specifications/User's commitment to acceptance test/User's experience in developing systems/ User's business experience/User's understanding level for design content/Clarity of role assignments and each organization's responsibilities between user and development staff
Requirement levels	Clarity level of requirement specifications/Requirement level (reliability/usability/performance and efficiency/portability/maintainability/ security) /Legal regulation
Development staff's skill levels	Project manager's (PM's) skill level
	Staff's skill levels (business experience/ analysis and design experience/ experience with languages and tools/ experience with development platform)
	Test team's skill levels and size

* "User" means individual or organization that uses the software system developed.

Table 5. Missing value ratios for qualitative variables

Missing value ratio	Number of qualitative variables
0.5 $\sim$ less than 0.6	7
0.6 $\sim$ less than 0.7	14
0.7 $\sim$ less than 0.8	11
0.8 $\sim$ less than 0.9	6
Total	38

3. Data analysis methods

3.1. Data pre-processing

(1) Transformation of quantitative variables

Histograms for the quantitative variables for software projects generally have an asymmetric distribution with a long tail on the right. The data should be transformed so that the distribution is more normal since statistical theories are generally based on a normal distribution. Since the hypothesis of a normal distribution for software project data that are logarithmically transformed is not usually rejected, all the quantitative variables, including that for productivity, were transformed before analysis.

(2) Merging of levels of qualitative data

Among the qualitative variables, some, such as project management tool usage have only two categories: "yes" and "no". Most of the qualitative variables, such as PM skill level have more than two categories, or levels. Twenty nine of the 39 candidate qualitative predictor variables in Table 4

Table 6. Number of qualitative values each project has

Number of qualitative values	Percentage (%)	Number of projects
38	100	38
35〜37	90〜99	36
31〜34	80〜89	7
27〜30	70〜79	2
23〜26	60〜69	40
19〜22	50〜59	32
16〜18	40〜49	14
12〜15	30〜39	17
8〜11	20〜29	40
4〜7	10〜19	83
1〜3	1〜9	100
0	0	114
Total		523

more than two levels. Before the analysis, the levels of variables with more than two levels were merged into two levels so that each level had more than or equal to 10 data values and 15% of the total amount of data values of the variable to satisfy the condition described in Section 2.2 (2). Similar levels were merged into one level. For example, four levels were typically merged into two levels by combining the upper two levels into one level and the lower two levels into another, or combining the top level or the lowest level into one level and the other three levels into the other level. The boundary was determined so that the numbers of two levels were as balanced as possible.

The levels of test team are not in monotonic order since they are defined as follows: level a; both the number of staff members and their skill levels are sufficient, level b; the number of staff members is insufficient but their skill levels are sufficient, level c; the number of staff members is sufficient but their skill levels are insufficient, and level d; both the skill level and the number of staff members are insufficient. Then, two combinations were made: levels a + b and c + d, and levels a + c and b + d. The former combination was focused on the number of staff members, and the latter on the skill level of staff members.

(3) Missing value

The missing values were not interpolated, and projects including missing values were omitted only when the variables related to these missing values were analyzed (pairwise deletion).

(4) Outliers

For the quantitative data, no "apparent" outliers could be found in each two-dimensional scatter graph. For the qualitative data, the conditions described in Section 3.1 (2) ensure the balance of the distribution.

3.2. Data analysis methods

3.2.1 Quantitative variables

Regression analysis was used to analyze the effect of the quantitative variables on effort. All combinations of the three predictor variables (size, number of test cases, and number of faults) were done. The coefficients of determination from the analysis were adjusted by taking into account the number of variables.

3.2.2. Qualitative variables

(1) One-dimensional analysis of variance

The Welch-adjusted analysis of variance was applied to each level pair of each variable regardless of their variance. If the P value was less than or equal to 5%, the variable was selected as a predictor variable.

If a variable satisfied the condition that both means of two levels were more than or less than the mean of all 523 projects, the variable was regarded as "biased" and reserved.

(2) Two-dimensional analysis of variance

A two-dimensional analysis of variance was used to detect the cross effects on productivity between each pair of selected predictor variables. Since the numbers in each category of the 2-by-2 cross table were different, the "simplified" two-dimensional analysis of variance [11] was used. A T-test (the Welch-adjusted analysis of variance) was applied to all combinations of the 2-by-2 levels of the cross table, that is, six pairs, to identify the combinations or categories that caused the most significant cross effects. If the number of signifcant combinations was less than three, that means no distinct level from others was not found, the pair of variables was omitted for further analysis.

4. Analysis results

4.1. Quantitative variables

Table 7 lists the results of (multiple) regression analysis for all combinations of three predictor variables. The number of projects decreases as the number of predictor variables increases. There are 523, 324, and 310 projects for each predictor variable, 324, 310, and 288 for each of the three pairs of predictor variables, and 288 for all three variables together, as shown in this table.

(1) One predictor variable

The predictor variable with the highest correlation to development effort is the size based on FP: 0.841. The adjusted coefficient of determination is 0.706. The 1.147 regression coefficient for size is a little greater than 1, mean-

Table 7. Regression analysis of quantitative variables that may affect development effort

Number of predictor variables	Regression coefficient				Multiple correlation coefficient	Adjusted coefficient of determination	Number of projects
	FP	Number of test cases	Number of faults	Constant			
1	1.147	-	-	0.566	0.841	0.706	523
	-	0.556	-	2.237	0.653	0.425	324
	-	-	0.686	2.575	0.706	0.497	310
2	0.893	0.230	-	0.637	0.855	0.730	324
	0.850	-	0.276	0.906	0.844	0.711	310
	-	0.278	0.483	2.108	0.772	0.594	288
3	0.746	0.193	0.182	0.797	0.868	0.750	288

ing that the increase in effort is proportionally slightly more than the increase in size. This result is consistent with the COCOMO formula.

The regression coefficients for the number of test cases and number of faults are 0.556 and 0.686, respectively, which means development efforts increases less than the increase in the number of test cases or number of faults.

(2) Two predictor variables together

The combination of predictor variables with the highest correlation to development effort is size plus number of test cases: 0.855. The corresponding adjusted coefficient of determination is 0.730, which means that about 73% of the development effort is determined by the software size and number of test cases in the test phases.

(3) All three predictor variables together

The multiple correlation coefficients of all three predictor variables are 0.868, and the coefficient of determination is 0.750 greater than that of two predictor variables together. This means that 75% of the development effort is determined by software size, number of test cases, and number of detected faults. Three predictor variables were not multicollinear, since variance inflation factor (VIF) values, the multi-collinear index, for three variables were 1.9, 2.0, and 2.5, respectively, and less than the criteria 10.0.

4.2. Qualitative variables

Of the 39 candidate variables, 20 had smaller P values than or equal to 5%. Of the 20 candidate variables, 7 were regarded as biased since both means of their levels were less than that of the productivity of all 523 projects: 0.988. Thirteen predictor variables were finally identified as factors affecting productivity alone (Table 8). Note that the P values of all three candidate predictor variables in the category of evaluation of plan were more than 40%.

Table 9 lists the 13 qualitative predictor variables, or factors, that affect productivity, together with the content of each level, number of projects, mean and variance of productivity, P values, coefficient of determination, productivity of the “typical project” of each level of the predictor

Table 8. Cross table of significant level and biased or unbiased predictor variables

	Unbiased	Biased	Total
Significant at 5%	13	7	20
Not significant	2	17	19
Total	15	24	39

variables, and productivity ratio of a typical project. A typical project is defined as a project whose size, effort, and productivity are the inverse logarithmic transformation of their means in the logarithmic scale.

(1) Overall project

- Of the seven candidate predictor variables in terms of the overall project, clarity of role assignments and each staff member’s responsibilities, clarity of objectives and priorities, working space and quality assurance system in the FD phase were significant at 1%.
- The productivity increases as clarity of role assignments and each staff member’s responsibilities, or clarity of objectives and priorities is very high, working space is broad enough, or quality assurance in the FD phase is performed by project members.
- The coefficient of determinations of clarity of objectives and priorities, and quality assurance system in the FD phase was more than 10%, while the others were more than 5%.
- The productivity ratios of the typical projects of all four predictor variables were more than 1.5. In particular, those of clarity of objectives and priorities and quality assurance system in the FD phase were more than 1.8.

(2) Tool usage

- Of the eight candidate predictor variables in terms of tool usage, project management tool usage, document-generation tool usage, and debug and test tool usage were significant at 1%, and similar project usage and development framework usage were significant at 5%.
- The productivity of a project in which a document-generation tool or development framework was used was

Table 9. Analysis of variance of qualitative variables that affect productivity and productivity ratio of typical projects

Category	Variable	Level	Number of projects	Productivity*		P value (upper)	Typical project	
				Mean (A)	Variance	Coef. of det.** (lower)	Produc-tivity *** (B)	Produc-tivity ratio
Overall project	Clarity of role assignments and each staff member's responsibilities	Very clear	84	0.820	0.147	0.0%	6.61	1.71
		Fairly clear + Little clear + Unclear	130	1.053	0.151	7.7%	11.31	
	Clarity of objectives and priorities	Very clear	70	0.754	0.128	0.0%	5.67	1.88
		Fairly clear + Little clear + Unclear	121	1.029	0.152	10.6%	10.68	
	Working space****	Levels: a + b (broad)	89	0.798	0.118	0.5%	6.28	1.56
		Levels: c + d (narrow)	66	0.991	0.208	5.0%	9.80	
	Quality Assurance system in FD phase	Done by project members	125	0.985	0.136	0.0%	9.65	1.84
		Done by quality assurance staff	59	1.249	0.121	10.0%	17.73	
Tool usage	Similar project	Usage	54	1.009	0.165	3.3%	10.21	1.47
		No usage	57	0.843	0.165	3.2%	6.96	
	Project management tool	Usage	111	1.004	0.181	0.0%	10.09	1.63
		No usage	64	0.791	0.110	5.8%	6.19	
	Document-generation tool	Usage	60	0.653	0.109	0.0%	4.50	2.21
		No usage	93	0.998	0.133	18.2%	9.95	
	Debug and test tool	Usage	52	1.003	0.208	0.8%	10.07	1.58
		No usage	99	0.806	0.126	4.8%	6.39	
	Development framework	Usage	91	0.923	0.158	1.8%	8.37	1.40
		No usage	75	1.070	0.156	2.8%	11.75	
User's skill levels and commitment	User's commitment to defining requirement specifications	Sufficient commitment + Fair commitment	132	0.917	0.162	2.6%	8.27	1.34
		Insufficient commitment + No commitment	81	1.043	0.155	1.9%	11.05	
Requirement levels	Requirement level for reliability	Extremely high + High	81	1.016	0.194	0.0%	10.38	1.85
		Medium + Low	87	0.750	0.101	10.4%	5.62	
	Requirement level for security	Extremely high + High	64	1.128	0.158	0.0%	13.43	2.85
		Medium + Low	89	0.672	0.074	31.5%	4.70	
Development staff's skill levels	PM's skill level*****	Levels 6 and 7 + Level 5 (high level)	58	1.088	0.195	0.0%	12.25	1.81
		Level 4 + Level 3 (low level)	108	0.831	0.140	8.2%	6.77	

*Log (Effort (person-hours)/FP)

**Coefficients of determination

***(B) is inverse transformation of (A). Unit is "Effort (person-hours) /FP".

****Definition of each level is shown in Annex.

*****Categorized in accordance with IT skill standard defined by METI.

higher than that of a project in which they were not used. The productivity of a project in which a project management tool, debug and test tool, or similar project was used was lower than that of a project in which they were not used.
- The coefficient of determination of document-generation tool usage and project management tool usage were 18.2 and 5.8%, respectively.
- The productivity ratio of the typical projects of document-generation tool usage, project management tool usage, and debug and test tool usage were 2.21, 1.63, and 1.58, respectively, while those of the others were less than 1.5.
(3) User's skill levels and commitment
- Of the six candidate predictor variables in terms of user's skill levels and commitment, only user's commitment to defining requirement specifications was significant at 5%.
- The productivity increases when the level of the user's commitment to defining the requirement specifications is high.
- The coefficient of determinations was less than 5%, and the productivity ratio of the typical projects was less than 1.5.
(4) Requirement levels
- Of the eight candidate predictor variables in terms of requirement levels, requirement level for reliability and requirement level for security were significant at 1%.
- The productivity decreases when requirement level for reliability or requirement level for security is high.
- The coefficient of determinations of requirement level for security was the highest: 31.5%, while that of requirement level for reliability was higher, 10.4%, than that of most predictor variables in other categories. Note that the sum of all coefficients of determination is over 1 since each qualitative variable was analyzed individually so that its coefficient of determination was overestimated.
- The productivity ratio of the typical projects of requirement level for security was 2.85, the largest among all predictor variables. The productivity ratio of requirement level for reliability for typical projects was also comparatively high: 1.85.
(5) Development staff's skill levels
- Of the seven candidate predictor variables in terms of development staff's skill levels, only PM skill level is significant at 1%.
- Productivity decreases when the PM skill level is high.
- The coefficient of determination and productivity ratio of the typical projects were 8.2% and 1.81, respectively.

4.3. Results of two-dimensional analysis of variance

Two-dimensional analysis of variance was applied for all combinations of the 13 predictor variables listed in Table 9. Of the 78 possible combinations, 17 had significant cross effects. Of the 17 combinations, 3 did not satisfy condition 2) described in Section 2.2 (2), 3 were regarded as biased, and 1 did not satisfy the condition that at least 3 of 6 pairs was significant for one-dimensional analysis of variance. Ten combinations were finally identified as those with cross effects.

The results for the synergy effect are listed in Table 10. All synergy effects work to lower productivity.
- The productivity of projects developing software with a high requirement level for security was lower than that with a low requirement level for security by 3.48 times when the working space was narrow, by 3.36 times when a development framework was not used, or by 3.06 times when the clarity of role assignment and each staff member's responsibility was not very clear.
- The productivity of projects developing software with a high requirement level for reliability was lower than that with a low requirement level for reliability by 3.07 times when the clarity of objectives and priority was not very clear, by 2.71 times when the clarity of role assignments and each staff member's responsibilities was not very clear, or by 2.22 times when a similar project was used.
- The productivity of projects developing software with a narrow working space was lower than that with a broad working space by 3.09 times when a development framework was not used, or by 2.35 times when the clarity of objectives and priorities was not very clear.
- The productivity of projects developing software using a project management tool was lower than that without it by 2.48 times when a document-generation tool was not used, or by 1.98 times when the clarity of objectives and priorities was not very clear.

5. Discussions

5.1. Quantitative variables

5.1.1 Multiple regression analysis

In accordance with the results presented in Section 4.1, the development effort can be predicted using

$$\begin{aligned}\log E &= 0.746\log S + 0.193\log T + 0.182\log B \\ &\quad +0.797 \\ &= 1.121\log S + 0.193\log(T/S) \\ &\quad +0.182\log(B/S) + 0.797, \qquad (1)\end{aligned}$$

where E denotes development effort (person-hours), S denotes software size (FP), T denotes the number of test cases and B denotes the number of faults.

We can transform Equation (1) into

$$E = 6.26S^{1.121}(T/S)^{0.193}(B/S)^{0.182}. \qquad (2)$$

Table 10. Results of two-dimensional analysis of variance

Combinations that cause synergy				Number of projects	Productivity*		P value	Productivity ratio**
Variable	Level	Variable	Level		Mean	Variance		
Requirement level for security	Extremely high + High	Working space***	Levels: c + d (Narrow)	34	1.264	0.146	0.1%	3.48
				110	0.722	0.086		
		Development framework	No usage	30	1.291	0.091	3.1%	3.36
				77	0.765	0.101		
		Clarity of role assignments and each staff member's responsibilities	Fairly clear + Little clear + Unclear	41	1.210	0.130	4.6%	3.06
				109	0.724	0.092		
Requirement level for reliability	Extremely high + High	Clarity of objectives and priorities	Fairly clear + Little clear + Unclear	45	1.197	0.123	0.0%	3.07
				108	0.711	0.098		
		Clarity of role assignments and each staff member's responsibilities	Fairly clear + Little clear + Unclear	42	1.179	0.165	2.1%	2.71
				111	0.746	0.115		
		Similar project	Usage	17	1.091	0.176	0.5%	2.22
				70	0.744	0.114		
Working space***	Levels: c + d (Narrow)	Development framework	No usage	25	1.299	0.125	0.7%	3.09
				80	0.809	0.132		
		Clarity of objectives and priorities	Fairly clear + Little clear + Unclear	47	1.138	0.183	0.4%	2.35
				108	0.768	0.107		
Project management tool	Usage	Document-generation tool	No usage	50	1.124	0.117	0.2%	2.48
				100	0.729	0.106		
		Clarity of objectives and priorities	Fairly clear + Little clear + Unclear	46	1.048	0.138	4.9%	1.98
				98	0.750	0.122		

*Log (Effort (person-hours)/FP). Upper: combinations that cause synergy. Lower: others.
**Using typical projects values that are computed by inversely transforming mean values on logarithmic scale.
***Definition of each level is shown in Annex.

Equation (2) shows that effort is proportional to the 1.121 times the power of size, 0.193 times the power of test case density, and 0.182 times the power of fault density. The 1.12 times the power of size is the same as shown in the average model (semidetached model) of the COCOMO formula, although SLOC was used as the size measure instead of FP.

By subtracting $\log S$ from both sides of Equation (1), we can transform it into

$$\log{(E/S)} = 0.121 \log S + 0.193 \log{(T/S)} + 0.182 \log{(B/S)} + 0.797. \quad (3)$$

Equation (3) shows that productivity is proportional to the root of the eighth power of size, root of the fifth power of test case density, and root of the fifth power of fault density.

5.1.2 Effect of numbers of test cases and faults as predictor variables

As explained above, adding the number of test cases and number of faults as predictor variables to the commonly used software size predictor variable results in a multiple correlation coefficient of 0.868 and an adjusted coefficient of determination of 75.0%, which is larger than 70.6%.

Although the effect of adding these two variables is smaller than expected before analysis, these two variables may be fairly useful for the following reasons. The test case density is highly related to test efforts, and can be controlled by a project manager. The fault density, which also affects test efforts, cannot be directly controlled by a project manager and is unpredictable at the planning stage. However, as the project progresses, the quality of the product will become clearer; thus fault density will be predictable.

5.2. One-dimensional regression analysis of variance

The 13 candidates identified as factors affecting productivity in Section 4.2 are investigated from the following two questions.

a) What is the productivity ratio of typical projects of two groups in addressed?

b) Is the selected predictor variable appropriate for affecting productivity as determined from the literature and my experience?

Of the 13 ratios, 11 are less than 1.9, except for that of requirement level for security (2.85) and that of document-generation tool usage (2.21). In other words, most factors alone affect productivity of less than 2.0 except for these two variables described above.

(1) Overall project

The finding suggesting that the productivity of projects for which role assignments and each staff member's responsibilities or objectives and priority is very clearly defined or working space is broad enough is higher than otherwise is appropriate for this study since these circumstances make developers work effectively without physical stress or mental confusion.

The finding suggesting that the productivity of projects in which project members ensure the quality of the design specifications is higher than that in which the staff of the quality assurance team does is a little different from what is written in textbooks or reported in research papers. For example, it is said that less effort would be needed in the test phases and also in all phases when a quality assurance team ensures the design specifications than when project members do since design specifications ensured by the staff of a quality assurance team have generally fewer defects than those ensured by the project members. Analysis from the viewpoint of quality (or reliability) instead of productivity may lead to a different conclusion in which the quality of projects in which the staff of the quality assurance team ensures the quality of the design specifications is higher than that of project members. Therefore, this factor should not be considered as negative.

(2) Tool usage

The finding, suggesting that the productivity of projects in which either a document-generation tool or development framework is used is higher than otherwise, is reasonable. However, the finding, suggesting that the productivity of projects in which either a similar project, project managing tool, or debug and test tool is used is lower than otherwise, is an unexpected result. Further study is needed since the usage of these tools or similar project may contribute to improving reliability.

(3) User's skill levels and commitment

The finding, suggesting that the productivity of projects in which the user has insufficient or no commitment to defining the requirement specifications is lower than otherwise, is reasonable and should be a recognized axiom.

(4) Requirement levels

The finding, suggesting that the productivity of projects developing software with a high requirement level for security is lower than that of projects developing software with a low requirement level, is reasonable as software requiring a high security level would have many complex functions difficult to develop.

The finding, suggesting that the productivity of projects developing software with a high requirement level for reliability is lower than that of projects developing software with a low requirement level, is also reasonable.

(5) Development staff's skill levels

The finding, suggesting that the productivity of projects managed by a person with a high skill level was 1.81 times lower than that of projects managed by a person with a low skill level, is inappropriate. This may be due to a difference in the software size of the two projects. In fact, the software

Table 11. Difference in project features conducted by high and low skill PMs

	Project size in FP		Test case density		Fault density	
PM skill level	high	low	high	low	high	low
Mean*	3.114	2.882	0.303	0.047	-0.884	-0.932
Variance*	0.221	0.199	0.166	0.388	0.543	0.396
Number of projects	58	108	40	65	40	65
P value	0.3%		1.2%		18.2%	
Ratio of typical projects	1.71		1.80		1.12	

*Logarithmic scale

Table 12. Peason's Chi-squared Test between PM skill levels and three predictor variables

		User's commitment to defining requirement specifications		Requirement level for reliability		Requirement level for security	
		Sufficient commit-ment + Fair commitment	Insufficient commit-ment + No commitment	Extremely high + High	Medium + Low	Extremely high + High	Medium + Low
PM skill level	High	40	4	31	12	26	17
	Low	78	27	47	58	37	67
P value		3.9%		0.4%		1.0%	

size of projects managed by a person with a high skill level is more than 1.71 times larger than that of those managed by a person with a low skill level (Table 11). However, the difference in software size can compensate for the difference in their productivity by at most 1.07 times.

Further investigation to resolve this perplexing problem revealed that key differences between two categories in accordance with the PM skill level were test case density, as shown in Table 11, and requirements level for reliability and security, as shown in Table 12. These results indicate that a PM with a high skill level tends to manage a software project developing large-scale software with high requirement levels for reliability or security, which is difficult to develop. One of their duties is to run much more test cases per FP to maintain software quality than a PM with low skill level.

5.3. Two-dimensional analysis of variance

Of the 10 combinations, 5 productivity ratios were greater than 3, and 4 were between 2 and 3. These results indicate that most combinations of these predictor variables affect productivity more than the predictor variables do alone. In particular, the productivity of a project developing software whose requirement level for security was high in a narrow working space was 3.48 times lower than otherwise, or the productivity of a project developing software whose requirement level for security was high without a development framework was 3.36 times lower than otherwise. As the requirement level for security alone results in productivity ratio by 2.85, the synergy effect magnifies the productivity ratio caused by the requirement level alone by nearly 1.2 times.

For the requirement level for reliability, magnification of the synergy effect with clarity of objectives and priorities together was 1.66 times because the productivity ratio for the requirement level for reliability alone was 1.85 and that for the requirement level for reliability with clarity of objectives and priorities together was 3.07.

For narrow working space, magnification of the synergy effect with development framework was nearly double because the productivity ratio of working space alone was 1.56 and that of the combination of a narrow working space and no usage of develoment framework was 3.09.

Project management tool usage slightly magnifies the productivity ratio of document-generation tool usage alone, and clarity of objectives and priorities alone. The ratios 2.21, 1.88 were magnified to 2.48, 1.98 respectively, then the magnifications were 1.12, 1.05 times. The size of the software of projects using a project management tool was larger than that of projects without a project management tool, as shown in Table 13, which could be interpreted that the effect of document-generation tool usage alone, or clarity of objectives and priorities alone is magnified in large-scale software development.

All synergy effects greatly lower the productivity. The following combinations of synergy effects are the most important.

Table 13. Difference in FP between projects with and without project management tool

Level	Number of projects	Mean	Variance	P Value	Size ratio*
Usage	111	3.023	0.255	0.0%	2.01
No usage	64	2.720	0.174		

*Ratio of typical project values

- When developing software required for high security or for high reliability, role assignments and each staff member's responsibilities, and objectives and priorities should be very clear, and working space should be broad enough. Such a project has a high possibility of extremely lower productivity.
- To prevent lower productivity, the usage of a development framework is also important when developing software required for high security.

5.4 Comparison of factors to scale factors and cost drivers of COCOMO II

Table 14 compares the factors selected in this research to the scale factors and cost drivers introduced in COCOMO II. Note that the comparison is not rigorous but only provides suggestions to practitioners because the results for the qualitative variables in this paper are derived from one-dimensional or two-dimensional regression analysis of variance, which will be different from the results of multiple regression analysis. Also, note that the meanings of two sets of values, ratio of productivity, and productivity range shown in the Table 14, are different. The values could be used within the same set to compare the magnitude of the effect of each variable or each scale factor/cost driver, respectively.

The noticeable differences between the factors and the scale factors and cost drivers of COCOMO II are as follows.
- For the variables related to the category overall project in the SEC database, which are different from the scale factors and cost drivers in COCOMO II, four factors were found. Some of these factors may be more effective if they are used as scale factors in the formula as those of COCOMO II.
- In the category of tool usage, eight qualitative variables except for similar project are defined in the SEC database while only one aggregate cost driver for tool usage is introduced in COCOMO II. Of the eight variables, four were significant at 5% and not biased, two of them, that is, development framework and document-generation tool, increase productivity and the others decrease it.
- Six variables related to the category of user's skill levels and commitment are provided in the SEC database, while no cost drivers related to the category are found in COCOMO II. Four variables were significant at 10% including two biased variables, although only user's commitment to requirement specifications was selected as a factor affecting productivity.
- Although more detailed requirements related variables are specified in the SEC database than the cost drivers of COCOMO II, the requirement level for usability, requirement level for portability, and clarity level of requirement specifications are not significant even at 10%. The factors of requirement level for reliability, requirement level for security, and requirement level for performance and efficiency corresponded well to the cost drivers of required software reliability, product complexity, and time constraint in COCOMO II together with their effective levels of productivity, respectively.
- The variables in the category development staff's skill levels in the SEC database are similar to those of the cost drivers in COCOMO II. The distinct differences between the variables and cost drivers are PM skill level in the SEC database and personnel continuity in COCOMO II. The analysis results were not similar to the cost drivers in COCOMO II, such that only staff experience with development platform except for PM skill level was significant at 5%, while six cost drivers were identified in COCOMO II.

6. Summary

The SEC database keeps more than 3000 enterprise software projects with many more quantitative and qualitative variables than other databases open to the world. However, it seems difficult to effectively use this database for constructing cost models because of abundant missing values. As the first step, the factors that affect the productivity of developing enterprise software were clarified by analyzing the data for 523 projects. Several interesting results were obtained.

- Productivity was proportional to the root of the fifth power of test case density, and that of fault density in addition to the root of the eighth power of size.
- Thirteen predictor variables alone were identified. The most effective top four were requirement level for security, document-generation tool usage, clarity of objectives and priorities, and requirement level for reliability.
- The productivity ratios of typical projects of most factors were less than 2.0 except for two factors: 2.85 of requirement level for security and 2.21 of document-generation tool.
- The productivity of projects managed by a person with a high skill level was 1.81 times lower than that of projects managed by a person with a low skill level. One of the reasons was PMs with high skill level tended to manage

Table 14. Comparison of results derived from SEC database and cost drivers of COCOMO II

Category	Results derived from SEC database: Qualitative candidate predictor variable	Results derived from SEC database: Productivity ratio of typical projects*	Similarity**	COCOMO II: Scale factor and cost driver***	COCOMO II: Productivity range
Overall project	Quality Assurance system in FD phase	1.84	≠	Process Maturity †	1.43
	Clarity of role assignment and each staff member's responsibility	1.71		Architecture and Risk Resolution †	1.39
	Clear Objective and priority	1.88		Team Cohesion †	1.29
	Quantitative quality criteria for delivery	-		Development Flexibility †	1.26
	Introduction of new technology	++		-	
	Working space	1.56		Platform Volatility	1.49
	Noise conditions	#1.88		Multi-site Development	1.53
Tool usage	Similar project	1.47	∼	Precedentedness †	1.33
	Development framework	1.40	∼	Use of Software Tools	1.50
	Project management tool	1.63			
	Configuration control tool	#1.56			
	Design support tool	++			
	Document-generation tool	2.21			
	Debug and test tool	1.58			
	CASE tool****	-			
	Code generator	-			
User's skill levels or commitment	User's commitment to defining requirement specifications	1.34	-		
	User's commitment to acceptance test	#1.37			
	User's experience in developing systems	-			
	User's business experience	-			
	User's understanding level for design content	#1.51			
	Clarity of role assignment and each organization's responsibility between user and development staff	++			
Requirement levels	Requirement level for reliability	1.85	=	Required Software Reliability	1.54
	Requirement level for security	2.85	∼	Product Complexity	2.38
	Requirement level for performance and efficiency	#1.36	∼	Time Constraint	1.63
	Requirement level for usability	-	≠	Storage Constraint	1.46
	Requirement level for portability	-		Data Base Size	1.42
	Requirement level for maintainability	++		Develop for Reuse	1.31
	Legal regulation	#1.82		Required Development Schedule	1.43
	Clarity level of requirement specifications	-		Documentation Match to Life-cycle Needs	1.52
Development staff's skill levels or experience	PM skill level	1.81	≠	Analyst Capability	2.00
	Staff's analysis and design experience	-		Programmer Capability	1.76
	Staff's business experience	-	∼	Application Experience	1.51
	Staff's experience with development platform	#1.43	=	Platform Experience	1.40
	Staff's experience with languages and tools	-	=	Language and Tool Experience	1.43
	Test team's skill levels and size	-	≠	Personnel Continuity	1.59

Note that productivity ratios of typical projects and productivity range have different meanings.

* No mark is significant at 5%. "#" is significant at 5%, but biased. "++" is significant at 10%.

** "=" means " (Nearly) equal to", "∼" means "Similar to", and "≠" means "Does not correspond to".

*** † means scale factor in COCOMO II formula.

**** Omitted for analysis because number of projects for each level was extremely unbalanced.

software projects developing large-scale software with high requirement levels for reliability and security, which is difficult to develop. One of their duties is to run much more test cases per FP to maintain software quality than a PM with low skill level.

• Two-dimensional analysis of variance revealed 10 synergy effects. The following two cases were the most notable; the productivity of a project developing software required for high security in a narrow working space was 3.48 times lower than otherwise, and the productivity of a project developing software required for high security without a development framework was 3.36 times lower than otherwise.

The factors derived from the SEC database were compared to the cost drivers of COCOMO II.

• For overall project, four factors and two possible predictor variables were derived from the SEC database, while four scale factors and two cost drivers were identified, although they did not correspond to each other.

• Two factors related to tool usage specified in the SEC database were positive on productivity, but two were negative, although only one cost driver related to tool usage was identified as a positive factor in COCOMO II.

• The SEC database provides variables related to user's skill levels and commitment, and one factor and three possible predictor variables were identified, while no cost driver was found in COCOMO II.

• For development staff's skill levels, no variable except for PM skill level was identified as the factor, while five cost drivers were found in COCOMO II.

Acknowledgements : This research was performed at Tokai University with the support of IPA SEC based on the Liaison program of software engineering practice between industrial and academic areas supported by the Ministry of Economy, Trade and Industry (METI) of Japan. The author thanks the successive Directors of the SEC, SEC researchers, and members of the data analysis section of the Enterprise Software Committee.

Annex : Definitions of four levels of working space.
Level a: Sufficiently broad space closed for each individual.
Level b: Ordinary space for each individual and environment in which it is appropriate to concentrate.
Level c: Narrow open space, and environment in which it is difficult to concentrate.
Level d: Narrow open space with neither materials nor computers.

References

[1] Jϕrgensen, M, and Shepperd, M.: A Systematic Review of Software Development Cost Estimation Studies, IEEE Trans. Software Engineering, Vol.33, No.1, pp.33-53 (2007).
[2] Dejaeger, K., Verbeke, W., Martens, D., and Baesens, B.: Data Mining Techniques for Software Effort Estimation: A Comparative Study, IEEE Tr. Software Eng., Vol. 38, No. 2, pp. 375-397 (2012).
[3] Boehm, B. W.: Software Engineering Economics, p. 767, Prentice-Hall, Inc. (1981).
[4] B. W. Boehm et al.,: Software Cost Estimation with COCOMO II, p. 502, Prentice Hall PTR, (2000).
[5] Maxwell, K. D., Van Wassenhove, L. and Dutta, S.: Software Development Productivity of European Space, Military, and Industrial Applications, IEEE Trans. Software, Vol. 22, No. 10, pp. 706-718 (1996).
[6] Maxwell, K. D. and Forselius, P.: Benchmarking software Development Productivity, IEEE Software, Vol. 17, No. 1, pp. 80-88 (2000).
[7] Sentas, P., Angelis, L., Stamelos, I., and Bleris, G.: Software productivity and effort prediction with ordinal regression, Information and Software Technology, Vol. 47, pp. 17-29 (2005).
[8] Chulani, S., Boehm, B. W., and Steece, B: Bayesian Analysis of Empirical Software Engineering Cost Models, IEEE Trans. Software Eng, Vol. 25, No. 4, pp. 573-583 (1999).
[9] http://promisedata.googlecode.com.
[10] IPA SEC: White Paper 2012-2013 on Software Development Projects in Japan, IPA, (2012). (in Japanese).
[11] Hirotsu, C: Analysis of experimental data, p.432, Kyoritsu Publishing Corp. (1992). (in Japanese).