Revisiting Software Development Effort Estimation Based on Early Phase Development Activities

Masateru Tsunoda
Toyo University
Saitama, Japan
tsunoda@ieee.org

Koji Toda
Fukuoka Institute of Technology
Fukuoka, Japan
toda@fit.ac.jp

Kyohei Fushida
NTT DATA Corporation
Tokyo, Japan
fushidak@nttdata.co.jp

Yasutaka Kamei
Kyushu University
Fukuoka, Japan
kamei@ait.kyushu-u.ac.jp

Meiyappan Nagappan
Queen's University
Ontario, Canada
mei@cs.queensu.ca

Naoyasu Ubayashi
Kyushu University
Fukuoka, Japan
ubayashi@ait.kyushu-u.ac.jp

***Abstract*—Many research projects on software estimation use software size as a major explanatory variable. However, practitioners sometimes use the ratio of effort for early phase activities such as planning and requirement analysis, to the effort for the whole development phase of the software in order to estimate effort. In this paper, we focus on effort estimation based on the effort for early phase activities. The goal of the research is to examine the relationship of early phase effort and software size with software development effort. To achieve the goal, we built effort estimation models using early phase effort as an explanatory variable, and compared the estimation accuracies of these models to the effort estimation models based on software size. In addition, we built estimation models using both early phase effort and software size. In our experiment, we used ISBSG dataset, which was collected from software development companies, and regarded planning phase effort and requirement analysis effort as early phase effort. The result of the experiment showed that when both software size and sum of planning and requirement analysis phase effort were used as explanatory variables, the estimation accuracy was most improved (Average Balanced Relative Error was improved to 75.4% from 148.4%). Based on the result, we recommend that both early phase effort and software size be used as explanatory variables, because that combination showed the high accuracy, and did not have multicollinearity issues.**

***Index Terms*—Effort prediction, early phase effort, function point, estimation accuracy, linear regression analysis.**

I. Introduction

In large projects, schedule and cost management is indispensable, and estimation of the total development effort is the basis of such management. So, high accuracy effort estimation (small difference between estimated and actual effort) is needed. One of major estimation methods is statistical model based estimation. When effort is estimated using this method, the model is trained using a dataset collected on past software development projects. To achieve high accuracy estimation, many estimation models have been proposed [1][6][20].

Many research projects on software estimation use software size (e.g., number of function points) as a major explanatory variable when effort is estimated [1][6][20]. However, practitioners sometimes use the ratio of effort for early phase activities such as planning and requirement analysis, to the effort for the whole development phase of the software in order to estimate effort [10]. Fig. 1 illustrates the estimation method based on early phase effort. For example, when ratio of the requirement analysis phase to the whole development phase on an average is 25% on past projects, and effort of requirement analysis in an ongoing project is 40 person-months, the whole effort of the ongoing project is estimated to be 160 person-months.

Software engineering researchers however, fail to include the early phase effort in effort estimation models. In this paper, we focus on estimation methods based on effort of early phase activities. We evaluate the improvement in accuracy of this method over traditional effort estimation models (i.e, linear regression models using software size [2]). Yang et al. [23] pointed out that there are few researchers who have analyzed the distribution of software development phase. Yang et al. also showed that the variance of the ratio of early phase effort (planning and requirement analysis phase effort) to whole development effort is not large. The small variance suggests that when effort of early phase is used as an explanatory variable, and an effort estimation model is built based on linear regression analysis, estimation accuracy is expected to be comparatively high.

As far as we know, there is no research project that compares the accuracy of the estimation model based on software size with the estimation model using early phase effort. Hence, there is no guideline that illuminates which of these two variables should be used as an explanatory variable when building an estimation model. The goal of the research is to examine the effect of size and early phase effort in effort estimation models. So, we set three research questions as follows:

- **RQ1**: When an effort estimation model using software size or early phase effort is built, which explanatory variable shows higher accuracy?
- **RQ2**: When other explanatory variables such as platform type are added to the models on RQ1, which shows higher accuracy, the model based on software size, or early phase effort?
- **RQ3**: When both software size and early phase effort are used as explanatory variables, is estimation accuracy improved? Does multicollinearity arise by using them? (When multicollinearity arises, both variables should not be used together)

Below, Section II describes the dataset used in the analysis. Section III shows preliminary analysis, and Section IV shows the experiment of effort estimation based on early phase development activities. Section V discusses the experimental results. Section VI explains the related work, and Section VII concludes the paper.

II. Dataset

A. Selecting Projects

We used a dataset that was collected from software development organizations in 20 countries by ISBSG (International Software Benchmarking Standards Group) [9]. The version of the dataset used in the case study is Release 9. The dataset includes data (99 variables) collected from 1989 to 2004 on 3,026 projects. There are some missing values in the dataset though. The variables from the dataset that we used in our experiment are shown in Table I.

As shown in Fig. 2, we extracted projects to be used in our case study as follows. (a) To ensure reliability of the experiment, we selected 1,255 projects that satisfy conditions shown by Locan et al. (e.g., data quality rating is A or B, and software size was measured by IFPUG method) [15]. (b) Preliminary elimination of projects: We eliminated projects in which the effort difference is more than 10%. The effort difference is defined as:

- Effort difference = (1 - sum of effort of all phases) / total effort (if sum of effort < total effort).
- Effort difference = (1 - total effort) / sum of effort of all phases (if sum of effort ≥ total effort).

In the equations, the sum of effort of all phases is sum of each phase effort from planning to implementation phase (in implementation phase the software is released and installed). We allowed the 10% effort difference to increase the number of analyzed projects. In addition, we eliminated projects in which function point is zero. Therefore from the 1,255 projects, we selected 172 projects in which both effort and function points is regarded as correct. (c) Projects used in the analysis for RQ1 and RQ3: from the subset, we selected 118 projects that did not have any missing values in the variables: planning effort and requirement analysis effort. (d) Projects used in the analysis for RQ2 and RQ3: from the subset, we selected 70 projects that did not have any missing values in the variables: development type, development platform, and language type. We use these projects in RQ2 and the second part of RQ3.

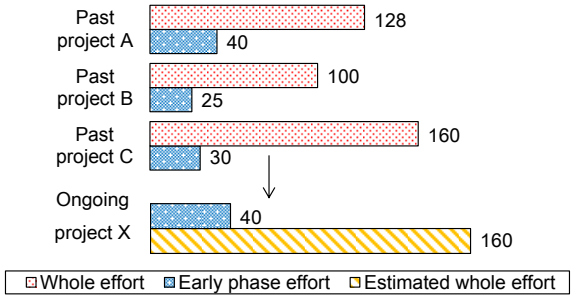

Fig. 1. Effort estimation based on early phase development activities

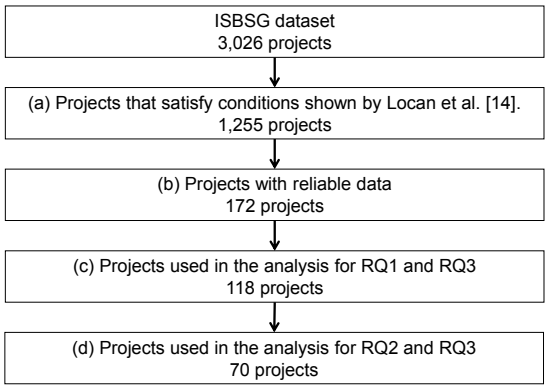

Fig. 2. Procedure of project extraction

B. Definition of Early Phase Effort

We want to use early phase effort as an explanatory variable in the estimation model for estimating the whole effort. However, early phase effort should be measured before it can be used in estimation models. We assumed that total effort is estimated after basic design, because software size (used as an explanatory variable) of selected projects was measured at that time (Software sizes of the projects were measured by IFPUG method, and IFPUG method requires to measure software size after basic design). Therefore, we defined that early phase effort can include basic design effort at most. In the ISBSG dataset, planning, requirement analysis and design effort is considered as early phase effort. However, we did not use design effort recorded in the ISBSG dataset, because it includes not only basic design effort, but also detail design effort.

Hence in our analysis we use planning-and-analysis effort (which as shown in Table I is the sum of planning effort and requirement analysis effort). Note that we do not use requirement analysis effort separately as an explanatory variable. Note also that we do not use planning effort and requirement analysis effort as explanatory variables separately because it causes multicollinearity. In addition to planning-and-analysis effort, we evaluate estimation models using planning effort, assuming effort (and software size) is estimated on very early phase.

III. Preliminary Analysis

A. Variance of Early Phase Ratio

When a mathematical model estimates effort, dataset collected from multiple or single organization is used [15].

TABLE I. Variables used in the Analysis

Variable	Scale	Description
FP	Ratio	Developed software size. Unadjusted function point measured by function point method.
Total effort	Ratio	Total effort spent to develop software.
Planning effort	Ratio	Effort spent to preliminary investigations, overall project planning, and so on.
Requirement analysis effort	Ratio	Effort spent to system analysis, requirement specify, architecture design/specification, and so on.
Planning-and-analysis effort	Ratio	Planning effort + requirement analysis effort
Planning ratio	Ratio	Planning effort / total effort
Planning-and-analysis ratio	Ratio	Planning-and-analysis effort / total effort
Duration	Ratio	Total project duration - project inactive duration
Development speed	Ratio	FP / duration
Productivity	Ratio	FP / total effort
Development type	Nominal	Development type of the project. The categories are new development, enhancement, and re-development
Development platform	Nominal	Platform in which developed software works. The categories are mainframe, midrange, and personal computer
Language type	Nominal	Programming language type used in the project. The categories are 2GL (second generation language), 3GL, 4GL, and application generator

Linear regression is one of common methods to build an estimation model [2]. Simple linear regression model is defined as:

$$y = b + ax. \quad (1)$$

In the equation, y is estimated effort (explained variable), x is an explanatory variable such as software size, a is a partial regression coefficient, and b is an intercept. To enhance the accuracy of the model, log transformation is often applied to both the explained and explanatory variable when building the model [13]. In the context of power law estimation [4], log transformation was applied for a single variable. In contrast, fitting at the doubly logarithmic plane is very common in software effort estimation research. For example, the traditional COCOMO model [1] was built based on that. When log transformation is applied and the estimation model is built based on simple linear regression analysis, the model is denoted as:

$$\log y = b + a \log x. \quad (2)$$

The equation is transformed as Eq. 4 via Eq. 3:

$$y = e^{b + a \log x}. \quad (3)$$

$$y = e^{b} x^{a}. \quad (4)$$

In the equations, e is the base of the natural logarithm. When c is defined as e^b, Eq. 4 is denoted as:

$$y = c x^{a}. \quad (5)$$

When software size is used as an explanatory variable, x is software size and c is regarded as a reciprocal of productivity. Productivity is a metric used to measure efficiency of finished software development, and it is defined as ratio of software size to total effort. However, in a simple linear regression model, c is a constant. Simply speaking, c is almost the same as average productivity of past projects, and it is inferred when building the model. If c is same on all projects and x (size) is given, y (effort) will be unique, because c is y / x (reciprocal of productivity). On the contrary, if c is not unique, y will be also not unique. So, small variance of productivity suggests small error. In contrast, if variance of an independent variable is extremely large, it is difficult to predict total effort by that variable alone. Note that the variance of a variable whose definition does not include y or x does not relate to estimation error.

Similarly, when early phase effort is used, x is early phase effort and c is regarded as a reciprocal of early phase ratio. So, c is almost the same as average of early phase ratio, and when variance in the distribution of early phase ratio is small, simple linear regression model using early phase effort is expected to have high accuracy.

To enhance the reliability of the experiment for RQ1, we compared the distribution of early phase ratio with the distribution of productivity to illuminate the cause of the difference of the estimation accuracy. Table II shows distributions of planning ratio, planning-and-analysis ratio, and productivity. We used ratio of third quartile to first quartile (Q_3 / Q_1), and ratio of Maximum value to minimum value (Maximum / Minimum), instead of variance. Q_3 / Q_1 was used to analyze the distribution, eliminating effects of outliers (Variance is affected by outliers). In contrast, Maximum / Minimum was used to analyze the distribution, focusing effects of outliers.

Considering Q_3 / Q_1 and Maximum / Minimum, distributions of planning ratio and planning-and-analysis ratio are narrower than productivity. This means planning ratio and planning-and-analysis ratio are not very different among projects, and suggests that the answer to RQ1 is "The accuracy of a simple linear regression model based on early phase effort is better than a model based on software size." The answer of RQ1 is effective for organizations that do not collect software development data in detail. Collecting data requires effort to some extent, and therefore some organizations do not have detailed data.

B. Relationships between Early Phase Ratio and Other Variables

If there are variables related to early phase ratio strongly, they should be used as explanatory variables, when a multiple linear regression model using early phase ratio (model in RQ2) is built. So, we applied bivariate analysis (variance explained in

TABLE II. DISTRIBUTIONS OF PLANNING RATIO, REQUIREMENT ANALYSIS RATIO, AND PRODUCTIVITY

Variable	Average	Minimum	Q_1	Median	Q_3	Maximum	Q_3 / Q_1	Maximum / Minimum
Planning ratio	0.090	0.005	0.039	0.071	0.124	0.434	3.2	86.8
Planning-and-analysis ratio	0.223	0.018	0.133	0.208	0.294	0.614	2.2	33.3
Productivity	0.197	0.003	0.057	0.127	0.288	1.044	5.1	348.0

TABLE III. RELATIONSHIPS BETWEEN NOMINAL SCALE VARIABLES AND EARLY PHASE RATIO

Variable		Development type	Development platform	Language type
Planning ratio	ω^2	0.01	0.03	-0.01
	p-value	0.18	0.10	0.52
Planning-and-analysis ratio	ω^2	-0.01	**0.08***	-0.01
	p-value	0.74	0.01	0.50

TABLE IV. RELATIONSHIPS BETWEEN RATIO SCALE VARIABLES AND EARLY PHASE RATIO

Variable		FP	Total effort	Productivity	Duration	Development speed
Planning ratio	ρ	-0.10	-0.17	0.12	**-0.28***	0.01
	p-value	0.25	0.05	0.18	0.00	0.88
Planning-and-analysis ratio	ρ	**-0.20***	**-0.20***	0.06	-0.18	-0.05
	p-value	0.03	0.03	0.51	0.08	0.62

ANOVA and correlation coefficient) to clarify the relationships between early phase ratio and other variables.

In addition, we analyzed relationships between early phase ratio and variables (total effort, productivity, duration, and development speed), which cannot be used as explanatory variables since they are not measured before estimation is done. For instance, development speed is not used as explanatory variable, because it uses project duration, and the duration is not settled before effort estimation. If there are such variables that affect early phase ratio, but we cannot use them as explanatory variables, then their influence should still be considered when applying the model using early phase effort. For example, if development speed (defined in Table I) strongly affects early phase ratio, and the speed is greatly different for each project in an organization, the model using early phase effort should not be used in the organization, because the accuracy of the model may be low. So, the relationship between such variables and the early phase ratio should be clarified to enhance reliability of the model.

Note that development platform and programming language do not seem to affect early phase ratio because planning and requirement analysis do not use programming language. However, if development platform and programming language affects other phase effort such as coding phase, then they indirectly affect early phase ratio. For instance, assume that followings:

- Project A: planning-and-analysis effort is 100 person-hour, and other phase effort is 300 person-hour, i.e., planning-and-analysis ratio is 25% of the total effort.
- Project B: planning-and-analysis effort is 100 person-hour, and other phase effort is 400 person-hour, i.e., planning-and-analysis ratio is 20% of total effort.
- If the programming language is the reason for the difference of other phase effort between the projects.

Then in this case, planning-and-analysis "ratio" is different between the projects, although planning-and-analysis "effort" is same. In contrast, there may be variables that affect early phase "effort" but not affect early phase "ratio." We did not analyze them because the analysis does not contribute to our research goal of examining the effect of early phase ratio in effort estimation models, directly.

To analyze relationships between nominal scale variables and early phase ratio, we used adjusted variance explained (ω^2) in ANOVA (analysis of variance). It is used to clarify the strength of the relationship between a nominal scale variable and a ratio scale variable. The range of the value is between 0 and 1, and larger value indicates the relationship is stronger. The value is calculated using the following equation [22].

$$\omega^2 = \frac{SSB - (k-1)MSE}{SST + MSE} \tag{6}$$

In the equation, *SSB* is the sum of squares between categories included in a nominal scale variable, *SST* is the sum of squares total, *MSE* is mean square error, and *k* is the number of categories.

Table III shows ω^2 and p-values between nominal scale variables and early phase ratio. In the table, * means the relationship were confirmed at significance level of 0.05. Although development platform and planning-and-analysis ratio (boldfaced) had significant relationship, it was not very strong. Other variables had no statistically significant relationship to early phase ratio variables.

To analyze relationships between ratio scale variables and early phase ratio, we applied Spearman's rank correlation coefficient (ρ). Software development dataset includes some large values (e.g., there are some projects whose software size is very large), and therefore some variables do not follow normal distribution. So, we used nonparametric method, instead of Pearson product-moment correlation coefficient. Table IV shows correlation coefficients and p-values between ratio scale variables and early phase ratio. In the table, * means the relationship was confirmed at a significance level of 0.05. Although FP and planning-and-analysis ratio, total effort and planning-and-analysis ratio, and duration and planning ratio had significant relationships (boldfaced), they were not strong. Other variables had no statistically significant relationship to early phase ratio variables.

On the dataset used in the experiment, there was no ratio scale and nominal scale variable that had a strong relationship to early phase ratio, based on the results of bivariate analyses. We did not find variables that should be used when an estimation model using early phase ratio is built, i.e., there is no variable that should be included as important explanatory variables when building effort estimation models. Additionally, total effort, productivity, duration, and development speed (variables that anyway cannot be measured in time to be used

in an effort estimation model) did not have strong relationships to early phase ratio.

IV. Estimation Based on Early Phase Effort

A. Procedure of Experiments

To answer RQ1 and RQ3, we built five types of estimation models using 118 projects, and evaluate the accuracies of them. The dependent variable is total development effort. Explanatory variables of the models are as follows:

- **Model I**: FP
- **Model II**: planning effort
- **Model III**: planning-and-analysis effort
- **Model IV**: planning effort and FP
- **Model V**: planning-and-analysis effort and FP

To answer RQ2 and RQ3, we added development type, development platform, and language type to the above models as candidates of explanatory variables, and built the models using 70 projects. They were used as explanatory variables in past study [15]. Based on the results of section III, they seem to be not very effective to enhance estimation accuracy of the model using early phase ratio. However, they may improve estimation accuracy of the model to some extent. So, we used the variables as candidates of explanatory variables on Model II and III. The variables should be used in Model I, IV, and V because there is a probability that they are effective when FP is used as explanatory variables.

We used linear regression analysis to build the models, and log transformation was applied to each ratio-scale variable. We applied variable selection based on AIC (Akaike's information criterion) when estimation models for RQ2 (using development type, development platform, and language type) were built. To check whether multicollinearity occurs or not, we used VIF (Variance Inflation Factor). When VIF of an explanatory variable is more than 2.5 or 3, it may be symptomatic of problematic multicolinearity in some situations, and when VIF is larger than 10, multicollinearity is considered to occur in the model. Nominal scale variables were transformed into a set of binary variables (one for each category of data present in the nominal variable), because linear regression analysis cannot handle nominal scale variables. Each binary variable was named as a category which the variable indicates.

We applied 5-fold cross validation to divide the dataset into fit datasets and test datasets. The fit datasets were used to build the models, and the test datasets were used to evaluate the models. We repeated 5-fold cross validation four times to increase number of evaluations (i.e., the evaluations were performed 20 times), because small number of evaluations causes type II error [8].

B. Evaluation Criteria

As evaluation criteria, we used average and median of ***AE*** (Absolute Error), ***MRE*** (Magnitude of Relative Error) [5], ***MER*** (Magnitude of Error Relative to the estimate) [12], and ***BRE*** (Balanced Relative Error) [17]. Especially, *MRE* is widely used to evaluate effort estimation accuracy [21]. We denoted average of *MRE* as *MMRE*, and median of *MRE* as *MdMRE*, for example. Small values of these evaluation criteria indicate that the accuracy of an effort estimation model is high.

When x denotes actual effort, and $\hat{x}$ denotes estimated effort, each criterion is calculated by the following equations:

$$AE = |x - \hat{x}| \tag{7}$$

$$MRE = \frac{|x - \hat{x}|}{x} \tag{8}$$

$$MER = \frac{|x - \hat{x}|}{\hat{x}} \tag{9}$$

$$BRE = \begin{cases} \dfrac{|x - \hat{x}|}{x}, x - \hat{x} \geq 0 \\ \dfrac{|x - \hat{x}|}{\hat{x}}, x - \hat{x} < 0 \end{cases} \tag{10}$$

Intuitively, *MRE* means error relative to actual effort, and *MER* means error relative to estimated effort. However, *MRE* and *MER* are imbalanced for underestimation and overestimation [3][14]. The maximum *MRE* is 1 even if an extreme underestimate occurs (For instance, when the actual effort is 1000 person-hour, and the estimated effort is 0 person-hour, *MRE* is 1). Similarly, maximum *MER* is smaller than 1 when an overestimate occurs. So, we gave weight to *BRE* that is balanced for them [18]. We did not use Pred(25) [5] which is sometimes used as an evaluation criterion, because Pred(25) is based on *MRE* and it has also a bias for evaluating under estimation.

We evaluated accuracies of models by differences in the evaluation criteria from a baseline model. Therefore, larger positive values mean estimation accuracies were improved from the baseline model, and negative values mean estimation accuracies got worse. The differences were tested statistically by Wilcoxon signed-rank test at a significance level of 0.05.

C. Models Using Software Size and Early Phase Effort Only

1) *Comparison of Estimation Accuracies of Models:* Table V shows accuracies of the five estimation models. In the table, first row indicates the accuracy of Model I (baseline model), and other rows indicate differences of accuracies between Model I, and other models (The accuracies were recorded as the average of the 20 evaluations). Larger values mean estimation accuracy was more improved, and negative values mean the accuracy was worse than Model I (model using FP). In the table, * denotes there was statistical difference in the corresponding evaluation criteria between Model I and the current model. About the estimation accuracies of the models, we observed the following:

- All criteria of Model II and III were better than Model I, and *MBRE* and *MdBRE* were statistically better than Model I. That is, estimation accuracy of a model using early phase effort was higher than software size in our case study.
- The accuracies of Model IV and V were better than Model II and III respectively, and compared with

TABLE V. MODELS USING SOFTWARE SIZE, AND EARLY PHASE EFFORT ONLY

Model		*MAE*	*MdAE*	*MMRE*	*MdMRE*	*MMER*	*MdMER*	*MBRE*	*MdBRE*
I (FP)		3725.6	1489.9	102.9%	60.3%	113.4%	62.7%	168.8%	104.1%
II (planning effort)	**Difference**	373.3*	122.3	28.2%*	5.6%	28.3%*	13.8%*	52.5%*	22.4%*
	p-value	0.00	0.28	0.00	0.12	0.03	0.00	0.00	0.01
III (planning-and-analysis effort)	**Difference.**	1214.1*	437.8*	49.5%*	17.6%*	53.5%*	20.0%*	91.6%*	45.1%*
	p-value	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
IV (planning effort and FP)	**Difference.**	805.2*	365.5*	37.1%*	8.6%*	40.3%*	17.4%*	69.8%*	33.7%*
	p-value	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
V (planning-and-analysis effort and FP)	**Difference**	1512.4*	681.3*	57.6%*	26.3%*	61.7%*	31.1%*	103.1%*	64.7%*
	p-value	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

TABLE VI. REGRESSION COEFFICIENTS OF MODELS USING SOFTWARE SIZE AND EARLY PHASE EFFORT ONLY

(A) MODEL I

Variable	*β*	VIF
FP	0.63	1

(B) MODEL II

Variable	*β*	VIF
Planning effort	0.77	1

(C) MODEL III

Variable	*β*	VIF
Planning-and-analysis effort	0.87	1

(D) MODEL IV

Variable	*β*	VIF
FP	0.33	1.34
Planning effort	**0.61**	1.34

(E) MODEL V

Variable	*β*	VIF
FP	0.25	1.37
Planning-and-analysis effort	**0.75**	1.37

Model I, the accuracies of Model IV and V were greatly improved, and all criteria of them were statistically better than Model I. Therefore, using both early phase effort and FP improved estimation accuracy in our case study.

- The accuracy of Model V was the best among all the models, and the accuracy of Model III was better than Model II and IV. Thus adding the analysis phase improved estimation accuracy in our case study.

2) *Evaluation of Effects of Variables and Multicollinearity:* Table VI shows standardized partial correlation coefficients (*β*) and VIF of models using software size and early phase effort only (They are the average of the 20 evaluations). About the built models, we observed the following:

- In Model IV and V (models using both software size and early phase effort), all VIFs were smaller than 2.5 in all 20 evaluations. So, multicollinearity did not arise when using both early phase effort and FP.
- In Model IV and V, the partial regression coefficients of early phase effort (boldfaced) were larger than software size. This means the effect of early phase effort was larger than software size, i.e., early phase effort was indispensable for the estimation models in our case study.

3) *Answers to Research Questions:* Based on the results, the answer to RQ1 is "Estimation accuracy of a model using early phase effort is higher than software size (when the model does not use other variables such as platform type)". The partial answer to RQ3 is "Using both early phase effort and FP improves estimation accuracy, and multicollinearity does not arise by using them, when the model does not use other variables such as platform type."

D. Models Using Software Size, Early Phase Effort, and Other Variables

1) *Evaluation of Adding Other Explanatory Variables:* To evaluate effect of adding other explanatory variables (development type, platform type, and language type), we compared the models without and with the variables. We set the models without the variables as baseline models. Table VII shows differences of estimation accuracies of the models. In the table, negative values mean estimation accuracies got worse when the other variables were added. Only the accuracy of Model I was improved, and that of other models got slightly worse. So, adding other explanatory variables did not improve estimation accuracy of the models using early phase effort.

2) *Comparison of Estimation Accuracies of Models:* Based on the above results, we set Model I with other variables as a baseline model, and compared the baseline model and other model without the other variables. Table VIII shows the comparison. About the estimation accuracies of the models, we observed the following in our case study:

- Six out of eight criteria of Model II (boldfaced) were worse than Model I. Estimation accuracy of a model using planning effort was lower than software size.
- In Model III, all criteria were better than Model I, and *MBRE* of it was statistically better. Therefore, the accuracy of a model using planning-and-analysis effort was higher than software size.
- The accuracies of Model IV and V were better than Model II and III respectively. In Model IV and V, all criteria were better than Model I, and *MBRE* of them were statistically better. That is, using both early phase effort and FP improved estimation accuracy.

3) *Evaluation of Effects of Variables and Multicollinearity:* Table IX shows standardized partial correlation coefficients (*β*) and VIF of models using software size, early phase effort, and other variables. In the table, # of times means the number

TABLE VII. COMPARISONS OF MODELS USING SOFTWARE SIZE AND EARLY PHASE EFFORT WITHOUT AND WITH OTHER VARIABLES [DEVELOPMENT TYPE, PLATFORM TYPE, AND LANGUAGE TYPE]

Model		*MAE*	*MdAE*	*MMRE*	*MdMRE*	*MMER*	*MdMER*	*MBRE*	*MdBRE*
I (FP without and with other variables)	**Difference**	-55.8	-15.0	22.7%*	0.6%	-4.3%	4.7%	16.2%	10.2%
	p-value	0.99	0.84	0.00	0.70	0.87	0.12	0.09	0.13
II (planning effort without and with other variables)	**Difference**	-112.8	-10.4	-2.6%	-3.6%	-8.1%*	-1.5%	-9.7%	-6.5%*
	p-value	0.01	0.72	0.34	0.09	0.00	0.34	0.05	0.02
III (planning-and-analysis effort without and with other variables)	**Difference.**	-122.9	-4.6	-1.8%	-2.3%	-4.6%*	0.4%	-5.3%	-0.2%
	p-value	0.52	0.52	0.65	0.59	0.02	0.65	0.05	0.78
IV (planning effort and FP without and with other variables)	**Difference.**	-212.4*	37.9	1.6%	0.0%	-3.5%	-2.9%	-2.1%	-3.1%
	p-value	0.02	0.23	0.43	0.99	0.90	0.26	0.78	0.62
V (planning-and-analysis effort and FP without and with other variables)	**Difference**	-166.4*	-124.0	-0.9%	-0.9%	-1.2%	-1.7%	-1.7%	-2.9%
	p-value	0.01	0.06	0.69	0.61	0.51	0.18	0.46	0.41

TABLE VIII. MODELS USING SOFTWARE SIZE WITH OTHER VARIABLES [DEVELOPMENT TYPE, PLATFORM TYPE, AND LANGUAGE TYPE], AND EARLY PHASE EFFORT WITHOUT OTHER VARIABLES

Model		*MAE*	*MdAE*	*MMRE*	*MdMRE*	*MMER*	*MdMER*	*MBRE*	*MdBRE*
I (FP, and other variables)		2966.6	1202.3	76.5%	48.7%	112.8%	48.8%	148.4%	72.5%
II (planning effort)	**Difference**	**-87.4**	**-81.9**	**-0.8%**	**-8.0%***	21.1%	**-0.3%**	22.9%	**-7.4%**
	p-value	0.55	0.39	0.93	0.04	0.62	0.81	0.81	0.28
III (planning-and-analysis effort)	**Difference.**	230.5	245.4*	23.9%*	0.8%	42.9%	8.8%*	62.0%*	11.8%
	p-value	0.15	0.04	0.00	0.60	0.05	0.01	0.00	0.07
IV (planning effort, FP)	**Difference.**	234.4*	193.9*	14.0%*	2.6%	37.8%	8.9%*	48.9%*	14.5%*
	p-value	0.04	0.01	0.00	0.57	0.11	0.02	0.01	0.10
V (planning-and-analysis effort, FP)	**Difference**	540.0*	527.9*	30.2%*	11.2%*	51.8%*	16.2%*	73.1%*	28.8%*
	p-value	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00

TABLE IX. REGRESSION COEFFICIENTS OF MODELS USING SOFTWARE SIZE, EARLY PHASE EFFORT, AND OTHER VARIABLES [DEVELOPMENT TYPE, PLATFORM TYPE, AND LANGUAGE TYPE]

(A) MODEL I

Variable	*β*	VIF	# of times
FP	0.71	1.07	20
3GL	0.27	1.14	20
MF	0.24	1.07	20
MR	0.14	1.82	1
New development	0.16	1.41	1

(B) MODEL II

Variable	*β*	VIF	# of times
Planning effort	0.74	1.08	20
3GL	0.17	1.05	12
MF	-0.18	1.20	8

(C) MODEL III

Variable	*β*	VIF	# of times
Planning-and-analysis effort	0.87	1.11	20
3GL	0.13	1.11	8
MF	-0.16	1.17	16
MR	-0.12	1.52	1

(D) MODEL IV

Variable	*β*	VIF	# of times
FP	0.44	1.33	20
Planning effort	**0.50**	1.40	20
3GL	0.19	1.06	20
MF	0.18	1.52	3

(E) MODEL V

Variable	*β*	VIF	# of times
Planning-and-analysis effort	**0.66**	1.60	20
FP	0.30	1.52	20
3GL	0.14	1.11	19
MF	0.14	1.75	1
New development	-0.09	1.25	2

of times the value was selected as an explanatory variable in the 20 evaluations. About the built models, we observed the following:

- VIF values were smaller than 2.5 in all 20 evaluations.
- The partial regression coefficient of planning-and-analysis effort was larger than software size in Model V, and that of planning effort was almost the same as software size in Model IV (boldfaced).

4) *Answers to Research Questions:* Based on the results, the answer to RQ2 is "Using other variables such as platform type slightly worsens estimation accuracy of models using early phase effort. Estimation accuracy of a model using planning effort is lower than a model based on software size using other variables, but the accuracy of a model using planning-and-analysis effort is higher than the model based on software size." Final answer to RQ3 is "Using both early phase effort and FP improves estimation accuracy, and multicollinearity does not arise by using them, regardless of the existence of other explanatory variables."

E. Comparison to Other Datasets

Same as preliminary analysis in section III, we compared early phase ratio of ISBSG dataset with other datasets, to enhance reliability of the experimental results. The datasets are ERA dataset and SEC dataset. We did not use CSBSG dataset used by Yang et al. [23] because some statistics about early phase ratio were not shown in it. The Economic Research Association collected the ERA dataset from 2001 to 2008 [7]. The dataset is collected from 268 software companies that are small to large companies in Japan. SEC dataset includes software development projects that were conducted in the 2000s in 23 large software companies [19]. In both datasets, planning phase ratio and requirement analysis ratio are not

TABLE X. DISTRIBUTIONS OF BASIC DESIGN PHASE RATIO AND PRODUCTIVITY ON ERA DATASET [7]

Development type	Variable	Number of projects	Average	Minimum	Q_1	Median	Q_3	Maximum	Q_3 / Q_1	Maximum / Minimum
New development	Basic design phase ratio	805	0.157	0.030	0.100	0.150	0.200	0.660	**2.0**	**22.0**
	Productivity	265	20.6	0.4	8.2	15.7	23.2	189.5	**2.8**	**473.8**
Enhancement	Basic design phase ratio	64	0.141	0.029	0.100	0.138	0.200	0.280	**2.0**	**9.7**
	Productivity	23	23.6	3.3	7.7	15.2	23.8	118.5	**3.1**	**35.9**

TABLE XI. DISTRIBUTIONS OF BASIC DESIGN PHASE RATIO AND PRODUCTIVITY ON SEC DATASET [19]

Development type	Variable	Number of projects	Average	Minimum	Q_1	Median	Q_3	Maximum	Q_3 / Q_1	Maximum / Minimum
New development	Basic design phase ratio	487	0.161	0.001	0.095	0.143	0.205	0.589	**2.2**	**589.0**
	Productivity	283	18.0	0.8	7.7	11.9	20.1	118.2	**2.6**	**140.7**
Enhancement	Basic design phase ratio	382	0.147	0.002	0.095	0.137	0.189	0.557	**2.0**	**278.5**
	Productivity	100	26.0	0.3	7.8	14.8	34.9	235.8	**4.5**	**873.2**

collected. Therefore we examine the "Basic design phase ratio" which is the ratio of basic design phase effort to sum of effort from basic design phase to system test phase. In ERA dataset, software size of most projects were measured by IFPUG method, and in SEC dataset, the size of all projects were measured by IFPUG method. Effort was measured as person-month. Note that each project data in ERA dataset and SEC dataset is not disclosed. However, distribution of variables (e.g., such as software size, programming language, and productivity) and relationships between the variables are shown in tables and figures in the books [7][19].

Distributions of basic design phase ratio and productivity in ERA dataset is shown in Table X, and that of SEC dataset is shown in Table XI. Q_3 / Q_1 and Maximum / Minimum of basic design phase ratio was smaller than that of productivity, except for Maximum / Minimum of new development in SEC dataset. The result suggests when basic design phase ratio is used as an explanatory variable of a simple linear regression model, the estimation accuracy of the model will be higher than the model using software size in ERA and SEC datasets. So, we conclude the answer of RQ1 is true on most datasets.

F. Effects of Adding Early Phase Effort

Adding independent variables enhances the fit of the model to the dataset. However, too many variables can cause overfitting. This will worsen estimation accuracy. Therefore to evaluate estimation models that consider a large number of independent variables, we use AIC (Akaike information criterion), which optimizes the fit and the number of independent variables. Smaller AIC means better model in both the fit and the number of variables, and adding a variable does not always improve AIC, when the improvement of the fit is small. As shown in table XII (AIC is the average of the 20 evaluations), Model IV had smaller AIC than model I and II, and model V had smaller AIC than model I and III. So, adding early phase effort is effective, even if we consider the additional variables.

TABLE XII. AICS OF BUILT MODELS

Model	Other variables	
	No	Yes
I (FP, and other variables)	154.3	120.2
II (planning effort)	141.0	119.1
III (planning-and-analysis effort)	113.8	91.2
IV (planning effort, FP)	125.1	99.5
V (planning-and-analysis effort, FP)	104.9	81.6

G. Guideline for Building Effort Estimation Model

Based on answers to research questions, we propose new guidelines for building effort estimation model as follows:

- (From the result of section IV. C. 1) If an organization does not collect project data in detail, it is preferable to build an effort estimation model that only uses early phase effort as an explanatory variable. It is expected to achieve reasonable estimation accuracy.
- (From the results of section IV. C. 1, IV. C. 2, IV. D. 2 and IV. D. 3) If software size is settled precisely by a method such as function point analysis before effort estimation, using both early phase effort and the size improves the accuracy without multicollinearity.
- (From the result of section IV. D. 1) In organizations that collect other project data in detail, it might not be preferable to use variables which we used (i.e., development type, development platform, and language type) as additional explanatory variables, in the estimation models using early phase effort, because that could possibly worsen the accuracy of the model.

V. DISCUSSION

A. Validity of Early Phase Effort

Although ISBSG defines that basic design effort and detail design effort should be a part of design effort, data suppliers might erroneously record them as requirement analysis effort or coding effort. Also in the ISBSG dataset the design effort was recorded in only four of 172 projects. Therefore, requirement analysis effort may include basic design effort and detail design effort (note that detail design effort is non-early-phase effort) in some cases.

In Table V and VII, estimation accuracy of Model III was higher than Model IV. This is because planning-and-analysis effort (which is used in Model III and is the sum of planning effort and requirement analysis effort) may include detail design effort, as stated above. So, this might enhance the accuracy of Model III (i.e., estimation accuracies of Model III

and V might not indicate the accuracy of a model using early phase effort properly).

In order to verify if the planning-and-analysis effort in the ISBSG dataset included detailed design effort too, we compared it with the data in another dataset – namely the CSBSG dataset analyzed by Yang et al. [23]. In the CSBSG dataset, the average of planning and requirement analysis phase was 16.1%, and that of (basic and detail) design phase was 14.9%. In the ISBSG dataset, the average sum of the planning and requirement analysis phase was 22.3%. Therefore the planning and requirement phase in the ISBSG dataset is much larger than the corresponding value in the CSBSG dataset, but much smaller than the sum of the planning and requirement, and design phases. So, if we assume phase distribution is almost the same between the datasets, the planning-and-analysis phase on ISBSG dataset does not seem to include detail design phase.

B. Influence of Reworks to Early Phase Effort

Although it is important to measure early phase effort accurately to enhance the estimation accuracy, it is not easy to measure it accurately. Although some reworks often occur on software during development, it is not easy to record their effort accurately, and that lessens the accuracy of early phase effort. Instead of measuring the effort of each phase precisely, it is good to focus on the total effort spent before a certain point of time in the development lifecycle, and regard it as the early phase effort. For example, total spent effort before project plan is made or a contract is made is defined as early phase effort (effort of reworks is ignored). This would standardize the measurement of effort, and enhance the estimation accuracy.

Planning effort and requirement analysis effort may include some errors, because some rework often occurs on software during development, and it makes the end of the phase somewhat obscure. However, total effort is more precise than them because the end of the project (when the project team breaks up) is clear. Therefore, although measurement error of planning effort and requirement analysis effort may lessen estimation accuracy of the models, it does not affect evaluation of the accuracy.

The estimation model using early phase effort assumes when using early phase effort is small, total effort is also small. There may be a project in which early phase effort is small due to insufficient work, and it increases effort on later phase. In this case, the assumption no longer fits the project, and estimation accuracy will worsen, when just using early phase effort in the estimation models.

C. Measuring Software Size

There are some methods which estimate or measure software size before basic design phase (e.g., NESMA functional size measurement method [11]). The answers of RQ1 and RQ2 may vary if software size and effort are estimated before basic design phase (We assumed they are estimated after basic design phase). However, we think the answer of RQ3 is useful if they are estimated before basic design phase. It is almost impossible to estimate software size precisely without requirement analysis, and the analysis requires some effort. That is, when software size is estimated or measured, some effort has been consumed. So, model using both estimated software size and effort can be built before basic design phase, and it is expected to show higher estimation accuracy.

Our result suggests that effort can be estimated without measuring or estimating software size. Our result would be effective in organizations that do not settle the size precisely before effort estimation (often because it is not very easy to measure or estimate the size before writing source code).

VI. Related Work

Some literature introduces simplified an estimation method using early phase effort and its ratio to the whole effort. For example, at the end of a phase, effort is estimated again using actual effort on the phase and average of the ratio of the phase to the whole phase, to confirm the progress of the project [10]. MacDonell et al. [16] investigated performance of models that use prior phase effort as an explanatory variable and estimate next phase effort, using 16 projects collected from a single software company. However, they did not use software size as an explanatory variable nor estimate total effort.

As far as we know, there is no literature that clarified which model shows higher estimation accuracy: a model using software size, or early phase effort. In addition, existing research projects does not clarify if using both software size and early phase effort improves estimation accuracy, and if multicollinearity arises. Yang et al. [23] pointed out that the distribution of development phase is often overlooked, although it is important for effort estimation. Major contribution of our research is not to propose new estimation method, but to confirm that using early phase effort, as an explanatory variable is expected to be effective in improving estimation accuracy, and to show new guidelines for building an estimation model. We think our guideline is easy to apply for practitioners, and its effectiveness is expected to be high.

VII. Conclusions

In this paper, we focused on an effort estimation method based on early phase effort and its ratio to the effort of the whole development process, and evaluated estimation accuracy of the model using early phase effort. Effort estimation based on early phase effort is sometimes used in practice, but it is not clear which shows higher estimation accuracy: a model based on software size or a model based on early phase effort. We set three research questions and answered them as follows:

- **RQ1**: When an effort estimation model using software size or early phase effort is built, which explanatory variable shows higher accuracy?
  Answer: (From the result of section IV. C. 1) Early phase effort showed higher accuracy than software size.
- **RQ2**: When other explanatory variables such as platform type are added to the models on RQ1, which shows higher accuracy, the model based on software size, or early phase effort?
  Answer: (From the results of section IV. D. 1 and IV. D. 2) Using other variables slightly worsens estimation

accuracy of models using early phase effort. A model using planning effort without the variables showed lower accuracy than a model using software size with the variables, but a model using planning-and-analysis effort without the variable showed higher accuracy than them.

- **RQ3**: When both software size and early phase effort are used as explanatory variables, is estimation accuracy improved? Does multicollinearity arise by using them?
  Answer: (From the results of section IV. C. 2 and IV. D. 3) Using both software size and early phase effort improved estimation accuracy, and multicollinearity does not arise by that, regardless of the existence of other explanatory variables.

Therefore, for organizations that do not measure software size, we recommend that effort be estimated using early phase effort, because it is expected to show relatively high accuracy. For organizations that measure software size, we recommend that both early phase effort and software size be used as explanatory variables, because they are expected to show high accuracy, and multicollinearity does not arise by using them.

ACKNOWLEDGMENT

This research was conducted as part of the Grant-in-Aid for Young Scientists (A) 24680003 by the Japan Society for the Promotion of Science.

REFERENCES

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

[2] L. Briand, T. Langley, and I. Wieczorek, "A replicated assessment and comparison of common software cost modeling techniques," In Proc. of International Conference on Software Engineering (ICSE), pp. 377–386, Limerick, Ireland, June 2000.

[3] C. Burgess, and M. Lefley, "Can genetic programming improve software effort estimation? A comparative evaluation," Journal of Information and Software Technology, Vol.43, No.14, pp.863–873, 2001.

[4] G. Concas, M. Marchesi, S. Pinna, and N. Serra, "Power-Laws in a Large Object-Oriented Software System," IEEE Trans. Software Eng., vol. 33, no. 10, pp. 687-708, Oct. 2007.

[5] S. Conte, H. Dunsmore, and V. Shen: Software Engineering, Metrics and Models, Benjamin/Cummings, 1986.

[6] A. Corazza, S. Martino, F. Ferrucci, C. Gravino, and E. Mendes, "Investigating the use of Support Vector Regression for web effort estimation," Empirical Software Engineering, Vol. 16, Issue 2, pp.211–243, 2011.

[7] Economic Research Institute, Economic Research Association: Analysis of Software Projects Data Repository, Economic Research Association, 2010. (in Japanese)

[8] A. Field, and G. Hole: How to design and report experiments, Sage Publications, 2003.

[9] International Software Benchmarking Standards Group (ISBSG): ISBSG Estimating: Benchmarking and research suite, ISBSG, 2004.

[10] International Software Benchmarking Standards Group (ISBSG): Practical ways to use the ISBSG data, http://www.isbsg.org/isbsgnew.nsf/WebPages/EB33DF6DB82ED1C5CA2576DC0081F046/$file/Practical%20Ways%20To%20Use%20The%20ISBSG%20Data.pdf

[11] International Organization for Standardization and International Electrotechnical Commission, ISO/IEC 24570:2005 Software engineering - NESMA functional size measurement method version 2.1 - Definitions and counting guidelines for the application of Function Point Analysis, 2005.

[12] B. Kitchenham, S. MacDonell, L. Pickard, and M. Shepperd, "What Accuracy Statistics Really Measure," In Proc. of IEE Software, Vol.148, No.3, pp.81–85, 2001.

[13] B. Kitchenham, and E. Mendes, "Why comparative effort prediction studies may be invalid," In Proc. of International Conference on Predictor Models in Software Engineering (PROMISE), No. 4 , p. 5, Vancouver, Canada, May 2009.

[14] C. Lokan, "What Should You Optimize When Building an Estimation Model?," In Proc. of International Software Metrics Symposium (METRICS), pp.34, Como, Italy, Sep. 2005.

[15] C. Lokan, and E. Mendes, "Cross-company and single-company effort models using the ISBSG Database: a further replicated study," In Proc. of the International Symposium on Empirical Software Engineering (ISESE), pp.75–84, Rio de Janeiro, Brazil, Sep. 2006.

[16] S. MacDonell, and M. Shepperd, "Using Prior-Phase Effort Records for Re-estimation During Software Projects," International Software Metrics Symposium (METRICS), pp.73, Sydney, Australia, Sep. 2003.

[17] Y. Miyazaki, M. Terakado, K. Ozaki, and H. Nozaki, "Robust Regression for Developing Software Estimation Models," Journal of Systems and Software, Vol.27, No.1, pp.3–16, 1994.

[18] K. Mølokken-Østvold, and M. Jørgensen, "A Comparison of Software Project Overruns-Flexible versus Sequential Development Models," IEEE Transactions on Software Engineering, Vol.31, No.9, pp.754–766, Sep. 2005.

[19] Software Engineering Center, Information-technology Promotion Agency: White Paper 2010-2011 on Software Development Projects in Japan, Information-technology Promotion Agency, 2010. (in Japanese)

[20] M. Shepperd, and C. Schofield, "Estimating Software Project Effort Using Analogies," IEEE Transactions on Software Engineering, Vol. 23, No. 11, pp. 736–743, November, 1997.

[21] F. Walkerden, and R. Jeffery, "An Empirical Study of Analogy-based Software Effort Estimation," Empirical Software Engineering, vol. 4, no. 2, pp. 135-158, 1999.

[22] B. Winer, D. Brown, and K. Michels, Statistical Principles in Experimental Design, McGraw-Hill, 1991.

[23] Y. Yang, M. He, M. Li, Q. Wang, and B. Boehm, "Phase Distribution of Software Development Effort," In Proc. of the International Symposium on Empirical Software Engineering and Measurement (ESEM), pp.384–392, Kaiserslautern, Germany, Oct. 2008.