Should Duration and Team Size be Used for Effort Estimation?

Takeshi Kakimoto[1]**, Masateru Tsunoda**[2]**, and Akito Monden**[3]

Abstract Project management activities such as scheduling and project progress management are important to avoid project failure. As a basis of project management, effort estimation plays a fundamental role. To estimate software development effort by mathematical models, variables which are fixed before the estimation are used as independent variables. Some studies used team size and project duration as independent variables. Although they are sometimes fixed because of the limitation of human resources or business schedule, they may change by the end of the project. For instance, when delivery is delayed, actual duration and estimated duration is different. So, although using team size and project duration may enhance estimation accuracy, the error may also lower the accuracy. To help practitioners to select independent variables, we analyzed whether team size and duration should be used or not, when we consider the error included in the team size and the duration. In the experiment, we assumed that duration and team size include errors when effort is estimated. To analyze influence of the errors, we add n% errors to duration and team size. As a result, using duration as an independent variable was not very effective in many cases. In contrast, using maximum team size as an independent variable was effective when the error rate is equal or less than 50%.

Key words software effort prediction, project management, productivity, estimation error

---

[1] Takeshi Kakimoto
Department of Electrical and Computer Engineering, National Institute of Technology, Kagawa College, Takamatsu, Japan. e-mail: kakimoto@t.kagawa-nct.ac.jp

[2] Masateru Tsunoda
Department of Informatics, Kindai University, Higashiosaka, Japan. e-mail: tsunoda@info.kindai.ac.jp

[3] Akito Monden
Graduate School of Natural Science and Technology, Okayama University, Okayama, Japan. e-mail: monden@okayama-u.ac.jp

1 Introduction

As recent software systems grow in size and complexity, project management activities such as staffing, scheduling and project progress management are becoming increasingly important to avoid project failure (cost overrun and/or delayed delivery). As a basis of project management, effort estimation plays a fundamental role; therefore, accurate effort estimation is vital to organization's profitability.

To date, various estimation models that use past projects' historical data have been proposed [2][23][25]. One of the most commonly used estimation models is a linear regression model, which represents the relationship between the dependent variable (i.e. effort) and independent variables such as functional size, architecture, programming language, and so on.

Analogy based estimation [23] is one of major estimation methods, and many proposals and case studies have been reported [8][9][20][27][31]. Analogy based estimation selects projects (neighborhood projects) which are similar to the estimated project from past project dataset, and estimates effort based on similar projects' effort. One of the advantages of analogy based estimation is that estimation results are comprehensible for estimators such as project managers [31], because they can confirm neighborhood projects used for estimation.

To estimate software development effort by mathematical models, variables which are fixed before the estimation are used as independent variables. Effort is estimated on the early phase of projects, i.e., after basic design phase. For example, architecture and programming language are fixed after the phase, and they are often used as independent variables of estimation models. In contrast, variables which are not fixed before the estimation cannot be used as the independent variables.

Some studies used team size [7][16] and project duration [1][12][13] as independent variables. However, they are not always fixed before the estimation. They are occasionally fixed after estimation. For example, when estimated effort is 9 person-months, duration is set as 9 and team size is set as 3. Sometimes, they are fixed because of the limitation of human resources or business schedule. But they may change by the end of the project. For instance, when delivery is delayed, actual duration (fixed after project is finished) and estimated duration (value input to the model) is different. Generally, estimation model is made based on the actual duration of past projects, and therefore input value (estimated effort) would include errors, as shown in Figure 1. Therefore, although using team size and project duration may enhance estimation accuracy, the error may also lower the accuracy.

The goal our study is to help practitioners to select independent variables when they build effort estimation models. So, we analyzed whether team size and duration should be used or not, when we consider the error included in the team size and the duration. To clarify the purpose of the analysis, we set following research questions:

- **RQ1**: Is duration effective to improve estimation accuracy?
- **RQ2**: Is team size effective to improve estimation accuracy?

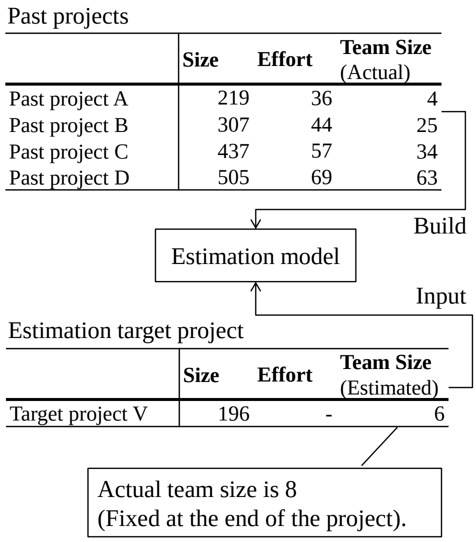

Figure 1 An example of the error included in team size

- **RQ3**: At what error rate is estimation accuracy negatively affected?

Section 2 explains effort estimation methods used in the experiment. Section 3 describes the experimental setting, and section 4 shows results of the experiment. Section 5 explains related work, and Section 6 concludes the paper.

2 Effort Estimation Methods

2.1 Multiple Linear Regression Model

The multiple linear regression model is widely used when estimating software development effort mathematically. The model is built based on ordinary least squares. When the effort is denoted as y, and independent variables such as software size are denoted as $x_1, x_2, \dots, x_k$ (k is the number of independent variables), y is explained as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon \quad (1)$$

In the equation, β_0 is an intercept, $\beta_1, \beta_2, \dots, \beta_k$ are partial regression coefficients, and ε is an error term. As a rule of thumb, to build a proper model using linear regression analysis, it is needed that the number of data points is five to ten times larger than the number of independent variables.

Table 1 Dataset used on analogy based effort estimation

	$Variable1$	$Variable2$	$\cdots$	$Variablej$	$\cdots$	$Variablel$
p_1	m_{11}	m_{12}	$\cdots$	m_{1j}	$\cdots$	m_{1l}
p_2	m_{21}	m_{22}	$\cdots$	m_{2j}	$\cdots$	m_{2l}
$\cdots$	$\cdots$	$\cdots$		$\cdots$		$\cdots$
p_i	m_{i1}	m_{i2}	$\cdots$	m_{ij}	$\cdots$	m_{il}
$\cdots$	$\cdots$	$\cdots$		$\cdots$		$\cdots$
p_k	m_{k1}	m_{k2}	$\cdots$	m_{kj}	$\cdots$	m_{kl}

When building the model, log-transformation is applied to enhance the accuracy of the model [10]. This is because the distributions of some variables such as effort and software size are log-normal distribution.

2.2 Analogy Based Estimation

The origin of analogy based estimation is CBR (case based reasoning), which is studied in artificial intelligence field. Shepperd et al. [23] applied CBR to software development effort estimation. CBR selects a case similar to current issue from accumulated past cases, and applies solution of the case to the issue. CBR assumes similar issues can be solved by similar solution. Analogy based estimation assumes neighborhood (similar) projects (For example, development size and used programming language is similar) have similar effort, and estimates effort based on neighborhood projects' effort. Although ready-made estimation models such as COCOMO [2] can make estimation without stored software project dataset, analogy based estimation cannot estimate without it. It is a weak point of analogy based estimation, but it can be overcome by using public dataset.

Analogy based estimation uses $k \times l$ matrix shown in Table I. In the matrix, p_i is *i-th* project, m_{ij} is *j-th* variable. That is, each row denotes a data point (i.e., a project), and each columns denotes a metric. We presume p_a is estimation target project, and $\hat{m}_{ab}$ is the estimated value of m_{ab}. Procedures of analogy based estimation consist of the three steps described below.

Step 1: Since each variable has different range of value, this step makes the ranges [0, 1]. The value m'_{ij}, normalized the value of m_{ij} is calculated by:

$$m'_{ij} = \frac{m_{ij} - min(m_j)}{max(m_j) - min(m_j)} \qquad (2)$$

In the equation, max(m_j) and min(m_j) denote the maximum and minimum value of m_j respectively. The equation is one of the commonly used methods to normalize the range of a value [26].

Step 2: To find projects which are similar to estimated project p_a (i.e., identifying neighborhood projects), similarity between p_a and other projects p_i is calculated. Variables of p_a and p_i are used as elements of vectors, and cosine of the vectors are regarded as similarity. Similarity $sim(p_a, p_i)$ between p_a and p_i is calculated by:

$$sim(p_a, p_i) = \frac{\sum_{j \in M_a \cap M_i} (m'_{aj} - avg(m'_j))(m'_{ij} - avg(m'_j))}{\sqrt{\sum_{j \in M_a \cap M_i} (m'_{aj} - avg(m'_j))} \sqrt{\sum_{j \in M_a \cap M_i} (m'_{ij} - avg(m'_j))}} \quad (3)$$

In the equation, M_a and M_i are set of variables measured in project p_a and p_i respectively. $avg(m', j)$ is average of *i-th* variable. The range of $sim(p_a, p_i)$ is [-1, 1].

Step 3: The estimated effort of project p_a is calculated by actual effort of k neighborhood projects. While average of neighborhood projects' effort is generally used, we adopt size adjustment method, which showed high estimation accuracy in some studies [9][20][31]. Estimated value $\hat{m}_{ab}$ is calculated by:

$$\hat{m}_{ab} = \frac{\sum_{i \in k-nearestProjects} (m_{ib} \times amp(p_a, p_i) \times sim(p_a, p_i))}{\sum_{i \in k-nearestProjects} sim(p_a, p_i)} \quad (4)$$

$$amp(p_a, p_i) = \frac{fp_a}{fp_i} \quad (5)$$

In the equation, fp_a and fp_i are software size of project p_a and p_i respectively. Size adjustment method assumes effort is s times (s is real number greater than 0) larger when software size is s times larger. The method adjusts effort of p_i based on ratio of target project's size fp_a and neighborhood project's size fp_i.

3 Experiment

3.1 Datasets

We used the ISBSG [6], Kitchenham [11], and Desharnais datasets [5]. Nominal scale variables were transformed into dummy variables (e.g. if the variable has n categories, it is transformed into n - 1 dummy variables). We removed dummy variables when the number of cases which correspond with the category was very small.

The ISBSG dataset is provided by the International Software Benchmark Standard Group (ISBSG), and it includes project data collected from software development companies in 20 countries [6]. The dataset (Release 9) includes 3026 projects that were carried out between 1989 and 2004, and 99 variables were recorded. The ISBSG dataset includes low quality project data (Data quality ratings are also included in the dataset).

We extracted projects based on the previous study [15] (Data quality rating is A or B, function point was recorded by the IFPUG method, and so on). Also, we ex-

Table 2 Variables of ISBSG Dataset

Variable	Scale	Description
FP	Ratio	Unadjusted function point
Effort	Ratio	Summary work effort (hour)
Duration	Ratio	Actual duration of project
Maximum team size	Ratio	Maximum number of personnel who engaged the project
Language type	Ratio	3GL (second-generation programming language), 4GL, and others
Development type	Nominal	New development, enhancement, and others
Development platform	Nominal	Mid range, main frame, and others

Table 3 Variables of Kitchenham Dataset

Variable	Scale	Description
FP	Ratio	Adjusted function point
Effort	Ratio	Actual development effort (hour)
Duration	Ratio	Actual duration of project
Development type	Nominal	Development, perfective, and others

Table 4 Variables of Desharnais Dataset

Variable	Scale	Description
FP	Ratio	Unadjusted function point
Effort	Ratio	Actual development effort (hour)
Duration	Ratio	Actual duration of project
Adjustment factor	Ratio	Adjustment factor of function point
TeamExp	Interval	Experience of team (measured in years)
ManagerExp	Interval	Experience of manager (measured in years)
Language	Nominal	type1, type2, and others

cluded projects that included missing values (listwise deletion). As a result, we used 196 projects. The variables used in our experiment are shown in Table 2. They are almost same as the previous study [15] except for duration and maximum team size.

The Kitchenham dataset includes 145 projects of a software development company, shown by Kitchenham et al. in their study [11]. We selected 135 projects that do not include missing values. Three variables shown in Table 3 were chosen as the independent variables, and inadequate variables for effort estimation (e.g. estimated effort by a project manager) were eliminated. Development type was transformed into dummy variables.

The Desharnais dataset includes 88 projects of 1980's, collected from a Canadian company by Desharnais [5]. The dataset is available at the PROMISE Repository [3]. We used 77 projects that do not have missing values. Variables shown in Table 4 were used as independent variables, and development year were not used. Also, the adjusted function point, the number of transactions, and the number of entities were not used to avoid multicollinearity. Programming language was transformed into dummy variables which reflects different development environments.

3.2 Evaluation criteria

To evaluate the accuracy of effort estimation, we used the conventional metrics such as AE (Absolute Error), *MRE* (Magnitude of Relative Error) [4], and *BRE* (Balanced Relative Error) [21]. Especially, *MRE* is widely used to evaluate the effort estimation accuracy [31].

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

$$AE = |x - \hat{x}| \tag{6}$$

$$MRE = \frac{|x - \hat{x}|}{x} \tag{7}$$

$$BRE = \begin{cases} \dfrac{|x - \hat{x}|}{x}, x - \hat{x} \geq 0 \\ \dfrac{|x - \hat{x}|}{\hat{x}}, x - \hat{x} < 0 \end{cases} \tag{8}$$

A lower value of each criterion indicates higher estimation accuracy. Intuitively, *MRE* means error relative to actual effort. However, *MRE* have biases for evaluating under estimation [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). So we employed *BRE* whose evaluation is not biased as is both *MRE* [22], and we evaluated the classified models based on mainly *BRE* (*MRE* were used for reference). We did not use Pred(25) [4] 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.

Learning data

	Size	Effort	Team Size (Actual)
Past project A	219	36	4
Past project C	437	57	34

Dataset

Test data (Team size error: 25%)

	Size	Effort	Team size (Including error)	Team Size (Actual)
Past project B	307	44	31.2	25
Past project D	505	69	50.5	63

Used as values of independent variables

Figure 2 Injecting errors into values of an independent variable

3.3 Procedure of Experiment

In the experiment, we assume that duration and maximum team size include errors when effort is estimated. This is because they are not fixed when effort is estimated (i.e., they are estimated values). At the end of the project, actual values of them may be different from the estimated values. To analyze influence of the errors, we add n% errors to duration and maximum team size. We set n as 0%, 25%, 50%, 100%, and 200%. The definition of the error rate is same as *BRE*.

Figure 2 is an example of the procedure. Dataset is divided into learning data and test data. Only team size on the test data includes the error. In the figure, team size on test data includes 25% errors. We generated new values of team size including the errors, and used it when effort is estimated.

We made the following models in the experiment, using analogy based estimation and multiple linear regression analysis.

A) Models without duration and maximum team size

B) Models with duration

C) Models with maximum team size

D) Models with duration and maximum team size

On model A, independent variables do not include duration and maximum team size. Model B includes duration as one of independent variables. In the same way, model C and D have independent variables. We call the model A as baseline, and evaluated other models with the baseline. Model C and D were made when ISBSG dataset is used. Since only ISBSG dataset includes maximum team size.

Table 5 Relationship to Effort and Productivity

Dataset	Variable	Effort	Productivity
Desharnais	Duration	0.57	-0.14
Kitchenham	Duration	0.57	-0.23
ISBSG	Duration	0.59	-0.17
ISBSG	Max. team size	0.68	-0.47

We evaluated accuracies of models by differences of criteria from a baseline model. Therefore, positive values mean estimation accuracies were improved from the baseline model, and negative values mean estimation accuracies got worse. 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.

Logarithmic transformation and variable selection was applied when multiple regression models were built. The number of neighborhoods was set as 5 when analogy based estimation was applied.

4 Results

4.1 Preliminary analysis

As preliminary analysis, we analyzed the relationship of duration and team size to effort and productivity. If the relationship is strong, using duration and team size as independent variables is expected to enhance estimation accuracy. Productivity was calculated by FP (function point) divided by effort. Strength of the relationship was evaluated using Spearman's rank correlation coefficient.

The result is shown in Table 5. The relationship between duration and productivity was weak on the three datasets, although the relationship between duration and effort was not weak. The result suggests that using duration as an independent variable is not very effective to enhance estimation accuracy. In contrast, strength of the relationship between maximum team size and productivity was moderate on ISBSG dataset. So, using maximum team size as an independent variable is expected to enhance estimation accuracy.

4.2 Estimation accuracy of analogy based estimation

Table 6 shows estimation accuracy of the models when analogy based estimation was used. In the table, top row of each dataset shows the accuracy of the model A (i.e., the baseline), and other rows do the difference from the baseline. Boldface in the table indicates the accuracy is improved using duration and maximum team size as independent variables.

Evaluation of model B (using duration): On Desharnais dataset, even the error rate is 0%, improvement of the accuracy was very small. Specifically, improvement of average *AE* and average *BRE* were very small, and median *AE* and median *BRE* got slightly worse. When the error rate of duration was equal or less than 50%, the negative influence to estimation accuracy was small. On Kitchenham dataset, average and median *BRE* were improved when the error rate was less than 100%. However, average and median *AE* got worse even the rate was 0%. On ISBSG dataset, estimation accuracy got worse on most cases. Therefore, when effort is estimated by analogy based estimation, using duration as an independent variable is not effective but sometimes negatively affects to estimation accuracy.

Evaluation of model C and D (using maximum team size): When maximum team size was used as an independent variable on ISBSG dataset (model C), it was effective to improve estimation accuracy. Except for median *MRE*, estimation accuracy was improved on most cases, when the error rate was equal or less than 50%. When both maximum team size and duration were used (model D), median *AE* and average *BRE* got worse. This would be because duration was negatively affected to the accuracy. So, using maximum team size as an independent variable is effective when the error rate is equal or less than 50%, and effort is estimated by analogy based estimation.

4.3 Estimation accuracy of multiple regression analysis

Table 7 shows estimation accuracy of the models when multiple regression analysis was used. The structure of the table is same as Table 6.

Evaluation of model B (using duration): On Desharnais dataset, average *AE*, *MRE*, and *BRE* were slightly improved, when the error rate was equal or less than 50%. In contrast, median *AE*, *MRE*, and *BRE* got worse. On Kitchenham dataset, estimation accuracy was improved when the error rate was equal or less than 50%. Especially, the improvement of average *BRE* was about 10%. On ISBSG dataset, using duration did not affect estimation accuracy very much, when the error rate was equal or less than 50%. Overall, using duration as independent variable did not negatively affected when multiple regression analysis was used, and sometimes positively affected when the error rate was equal or smaller than 50%.

Evaluation of model C and D (using maximum team size): When maximum team size was used as an independent variable on ISBSG dataset (model C), it was effective to improve estimation accuracy. Estimation accuracy was improved on

Table 6 Estimation Accuracy of Models When Using Analogy Based Estimation

Dataset	Variables	Error rate	Average *AE*	Median *AE*	Average *MRE*	Median *MRE*	Average *BRE*	Median *BRE*
Desharnais	Duration (Model B)	-	1884	1880	50%	49%	56%	46%
		0%	-38	**280**	-1%	**1%**	0%	0%
		25%	-9	**249**	-1%	**1%**	0%	**3%**
		50%	-72	**50**	-2%	**0%**	-1%	**2%**
		100%	-192	-221	-6%	-5%	-5%	-2%
		200%	-573	-681	-20%	-26%	-19%	-18%
Kitchenham	Duration (Model B)	-	1630	3221	79%	116%	100%	61%
		0%	-46	-566	-2%	-9%	**6%**	**13%**
		25%	-105	-673	-6%	-26%	**2%**	**11%**
		50%	-96	-732	0%	-6%	**6%**	**10%**
		100%	-403	-959	-22%	-25%	-17%	-10%
		200%	-813	-1384	-47%	-58%	-48%	-26%
ISBSG	Duration (Model B)	-	4444	6345	155%	247%	188%	104%
		0%	-781	-3519	-76%	-346%	-68%	**5%**
		25%	-463	-1346	-77%	-324%	-71%	-8%
		50%	-516	-1004	-69%	-251%	-64%	-6%
		100%	-579	-637	-78%	-244%	-74%	-6%
		200%	-561	-156	-76%	-201%	-74%	-22%
ISBSG	Team size (Model C)	-	4444	6345	155%	247%	188%	104%
		0%	**946**	**1346**	**9%**	-60%	**22%**	**26%**
		25%	**925**	**1356**	**4%**	-81%	**18%**	**26%**
		50%	**793**	**1183**	-7%	-98%	**6%**	**27%**
		100%	**425**	**883**	-24%	-142%	-13%	**12%**
		200%	-444	**202**	-78%	-241%	-71%	-29%
ISBSG	Duration Team size (Model D)	-	4444	6345	155%	247%	188%	104%
		0%	**426**	-502	-38%	-277%	-19%	**37%**
		25%	**209**	-371	-50%	-271%	-33%	**28%**
		50%	-88	-413	-63%	-285%	-51%	**19%**
		100%	-678	-753	-86%	-326%	-78%	-11%
		200%	-1919	-1879	-169%	-486%	-176%	-56%

most cases, when the error rate was equal or less than 50%. Also, when both max-

Table 7 Estimation Accuracy of Models When Using Multiple Linear Regression Analysis

Dataset	Variables	Error rate	Average *AE*	Median *AE*	Average *MRE*	Median *MRE*	Average *BRE*	Median *BRE*
Desharnais	Duration (Model B)	-	1652	1014	36%	29%	47%	35%
		0%	**7**	-56	**1%**	-2%	**1%**	-3%
		25%	**37**	-75	**1%**	-4%	**1%**	-4%
		50%	**22**	-59	**1%**	-5%	**1%**	-6%
		100%	-84	-166	-1%	-4%	-2%	-10%
		200%	-301	-370	-6%	-5%	-9%	-5%
Kitchenham	Duration (Model B)	-	1806	627	70%	40%	93%	52%
		0%	**203**	**18**	**5%**	**4%**	**12%**	**3%**
		25%	**185**	**38**	**6%**	**2%**	**12%**	**5%**
		50%	**138**	**76**	**5%**	**1%**	**9%**	**2%**
		100%	**20**	-51	-1%	-4%	-2%	-11%
		200%	-255	-247	-16%	-19%	-29%	-43%
ISBSG	Duration (Model B)	-	3496	1690	101%	55%	148%	92%
		0%	**3**	**133**	**4%**	-1%	0%	**6%**
		25%	**8**	**150**	**3%**	0%	-2%	0%
		50%	-3	**122**	**3%**	**2%**	-4%	**3%**
		100%	-33	**140**	**1%**	**2%**	-10%	**5%**
		200%	-143	**230**	-5%	0%	-21%	**1%**
ISBSG	Team size (Model C)	-	3496	1690	101%	55%	148%	92%
		0%	**636**	**570**	**12%**	**13%**	**31%**	**33%**
		25%	**434**	**494**	**5%**	**13%**	**22%**	**37%**
		50%	**134**	**322**	-5%	**6%**	**8%**	**27%**
		100%	-556	-57	-29%	-2%	-27%	**8%**
		200%	-1911	-735	-79%	-16%	-104%	-34%
ISBSG	Duration Team size (Model D)	-	3496	1690	101%	55%	148%	92%
		0%	**929**	**691**	**21%**	**20%**	**40%**	**46%**
		25%	**583**	**648**	**9%**	**14%**	**23%**	**36%**
		50%	-28	**324**	-9%	**5%**	-6%	**19%**
		100%	-1354	-386	-54%	-10%	-77%	-25%
		200%	-4030	-1993	-151%	-27%	-238%	-111%

imum team size and duration were used (model D), estimation accuracy was improved when the error rate is equal or less than 25%. When the error rate was 0%, the estimation accuracy of model D was better than the model C. However, then the rate is 25%, the accuracy was almost same. Therefore, using maximum team size as an independent variable is effective when the error rate is equal or less than 50%, but adding duration as an independent variable does not improve estimation accuracy unless the error rate is very small.

4.2 Summery of the results

Using duration as an independent variable (model B) was not very effective in many cases. Estimation accuracy was explicitly improved only when multiple regression analysis was used on Kitchenham dataset. So, the answer of RQ1 is "No."

In contrast, using maximum team size as an independent variable (model C) was effective when the error rate is not very large (equal or less than 50%). So, the answer of RQ2 is "Yes." To know the error rate, duration and maximum team size should be estimated and recorded, and we can calculate the rate when the data is accumulated.

When the error rate is equal or more than 100%, the estimation accuracy got worse in many cases. So, the answer of RQ3 is "100% and more." Overall, influence of duration, maximum team size and the error rate to estimation accuracy was not very different between analogy based estimation and multiple regression analysis. So, the influence would not be very different even when other estimation models are used.

5 Related Work

In our past studies, we focus on error included in independent variables such as difference between estimated team size and actual team size. Study [29] proposed an estimation method based on stratification of team size, and analyzed the influence of the error of team size. Also, study [28] proposed an estimation method based on productivity and proposed new method to absorb the influence of the error of the estimated productivity. However, study [29] used team size as a categorical variable, and not used as a ratio scale variable. Also, study [28] used productivity, but not used team size as an independent variable. Therefore, our past studies [28][29] did not clarify the effect of team size and duration to estimation accuracy.

There are many studies which analyzed the relationship between project attributes such as duration and productivity. For example, Maxwell et al. [17] and Premraj et al. [23] analyzed an influence of business sector for productivity, using

Finnish software development project dataset collected by Software Technology Transfer Finland (STTF). Lokan et al. [16] showed productivity by business sector using dataset of International Software Benchmarking Standards Group (ISBSG). In these studies, projects for manufacturing have the highest productivity, and projects for banking/Insurance have the lowest productivity.

Also, relationship of team size and duration to productivity was analyzed in some studies [18][30]. In the study [30], team size showed strong relationship to productivity, and duration was weak relationship to productivity. Dataset used in the study is Japanese cross-company dataset, and it is not ISBSG dataset. Therefore, our analysis result has external validity to some extent.

6 Conclusions

In this study, we evaluated the effect of using project duration and maximum team size as an independent variable on effort estimation models. We assume that duration and maximum team size include errors when effort is estimated. This is because they are not fixed on the point. To analyze influence of the errors, we add n% errors to duration and maximum team size. We set n as 0%, 25%, 50%, 100%, and 200%. We used ISBSG dataset, Kitchenham dataset, and Desharnais datasets in the experiment. To estimate effort, analogy based estimation and multiple linear regression analysis were used. Our findings include the followings:

- Using duration as an independent variable was not very effective in many cases.
- Using maximum team size as an independent variable was effective when the error rate is not very large (equal or less than 50%).
- When the error rate is equal or more than 100%, the estimation accuracy got worse in many cases.
- Influence of duration, maximum team size, and the error rate to estimation accuracy was not very different between analogy based estimation and multiple regression analysis.

The influence of maximum team size was evaluated only one dataset. To enhance the reliability of the results, we will analyze the influence in other dataset.

Acknowledgment

This research was partially supported by the Japan Ministry of Education, Science, Sports, and Culture [Grant-in-Aid for Scientific Research (C) (No. 16K00113)]

References

1. M. Azzeh, D. Neagu, and P. Cowling, "Fuzzy grey relational analysis for software effort estimation," Empirical Software Engineering, vol.15, no.1, pp.60-90, 2010.

2. B. Boehm, Software Engineering Economics, Prentice Hall, 1981.

3. G. Boetticher, T. Menzies, and T. Ostrand, PROMISE Repository of empirical software engineering data, West Virginia University, Department of Computer Science, 2007.

4. S. Conte, H. Dunsmore, and V. Shen, Software Engineering, Metrics and Models, Benjamin/Cummings, 1986.

5. J. Desharnais, Analyse Statistique de la Productivitie des Projets Informatique a Partie de la Technique des Point des Function, Master Thesis, University of Montreal, 1989.

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

7. R. Jeffery, M. Ruhe, and I. Wieczorek, "Using Public Domain Metrics To Estimate Software Development Effort," In Proceedings of the International Symposium on Software (METRICS), pp.16-27, 2001.

8. J. Keung, B. Kitchenham, and R. Jeffery, "Analogy-X: Providing Statistical Inference to Analogy-Based Software Cost Estimation," IEEE Transactions on Software Engineeing,vol.34, no.4, pp.471-484, 2008.

9. C. Kirsopp, E. Mendes, R. Premraj, and M. Shepperd, "An Empirical Analysis of Linear Adaptation Techniques for Case-Based Prediction," In Proc. of International Conference on Case-Based Reasoning, pp.231-245, 2003.

10. B. Kitchenham, and E. Mendes, "Why comparative effort prediction studies may be invalid," In Proceedings of International Conference on Predictor Models in Software Engineering (PROMISE), art.4, p.5, 2009.

11. B. Kitchenham, S. Pfleeger, B. McColl, and S. Eagan, "An Empirical Study of Maintenance and Development Estimation Accuracy," Journal of Systems and Software, vol. 64, no. 1, pp. 57–77, 2004.

12. Y. Li, M. Xie, and T. Goh, "A study of the non-linear adjustment for analogy based software cost estimation," Empirical Software Engineering, vol.14, no.6, pp.603-643, 2009.

13. J. Li, and G. Ruhe, "Analysis of attribute weighting heuristics for analogy-based software effort estimation method AQUA+," Empirical Software Engineering, vol.13, no.1, pp.63-96, 2008.

14. C. Lokan, “What Should You Optimize When Building an Estimation Model?” In Proceedings 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 Proceedings of the International Symposium on Empirical Software Engineering (ISESE), pp. 75–84, Sep. 2006.

16. C. Lokan, T. Wright, P. Hill, and M. Stringer, “Organizational Benchmarking Using the ISBSG Data Repository,” IEEE Software, vol.18, no.5, 2001, pp.26-32.

17. K. Maxwell, and P. Forselius, “Benchmarking Software Development Productivity,” IEEE Software, vol.17, no.1, pp.80-88, 2000.

18. K. Maxwell, L. Wassenhove, and S. Dutta, “Software Development Productivity of European Space, Military, and Industrial Applications,” IEEE Transactions on Software Engineering, vol.22, no.10, pp.706-718, 1996.

19. E. Mendes, S. Martino, F. Ferrucci, and C. Gravino, “Cross-company vs. single-company web effort models using the Tukutuku database: An extended study,” Journal of Systems and Software, vol.81, no.5, pp.673-690, 2008.

20. E. Mendes, N. Mosley, and S. Counsell, “A Replicated Assessment of the Use of Adaptation Rules to Improve Web Cost Estimation,” In Proc. of the International Symposium on Empirical Software Engineering (ISESE), pp.100-109, 2003.

21. 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.

22. K. Mølokken-Østvold, and M. Jørgensen, “A Comparison of Software Project Overruns-Flexible versus Sequential Development Models,” IEEE Transaction on Software Engineering, vol. 31, no. 9, pp. 754–766, 2005.

23. R. Premraj, M. Shepperd, B. Kitchenham, and P. Forselius, “An Empirical Analysis of Software Productivity over Time,” In Proceedings of International Software Metrics Symposium (METRICS), pp.37, 2005.

24. M. Shepperd, and C. Schofield, “Estimating software project effort using analogies,” IEEE Transaction on Software Engineering, vol. 23, no. 12, pp. 736–743, 1997.

25. K. Srinivasan, and D. Fisher, “Machine learning approaches to estimating software development effort,” IEEE Transaction on Software Engineering, vol. 21, no. 2, pp. 126–137, 1995.

26. K. Strike, K. Eman, and N. Madhavji, "Software Cost Estimation with Incomplete Data," IEEE Transactions on Software Engineering, vol.27, no.10, pp.890-908, 2001.

27. A. Tosun, B. Turhan, and A. Bener, "Feature weighting heuristics for analogy-based effort estimation models," Expert Systems with Applications, vol.36, no.7, pp.10325-10333, 2009.

28. M. Tsunoda, A. Monden, J. Keung, and K. Matsumoto, "Incorporating Expert Judgment into Regression Models of Software Effort Estimation," In Proceedings of Asia-Pacific Software Engineering Conference (APSEC), pp.374-379, 2012.

29. M. Tsunoda, A. Monden, K. Matsumoto, and A. Takahashi, "Software development effort estimation models stratified by productivity factors," SEC journal, pp.58-67, 2009. (in Japanese)

30. M. Tsunoda, A. Monden, H. Yadohisa, N. Kikuchi, and K. Matsumoto, "Software Development Productivity of Japanese Enterprise Applications," Information Technology and Management, vol.10, no.4, pp.193-205, 2009.

31. 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.