Benchmarking Software Maintenance Based on Working Time

Masateru Tsunoda
Nara Institute of Science and Technology
Nara, Japan
Department of Informatics, Kindai University
Osaka, Japan
tsunoda@info.kindai.ac.jp

Akito Monden
Graduate School of Natural Science and Technology
Okayama University
Okayama, Japan
monden@okayama-u.ac.jp

Kenichi Matsumoto
Graduate School of Information Science
Nara Institute of Science and Technology
Nara, Japan
matumoto@is.naist.jp

Sawako Ohiwa, Tomoki Oshino
Economic Research Institute
Economic Research Association
Tokyo, Japan
{er421, er352}@zai-keicho.or.jp

***Abstract*—Software maintenance is an important activity on the software lifecycle. Software maintenance does not mean only removing faults found after software release. Software needs extensions or modifications of its functions due to changes in a business environment, and software maintenance also indicates them. In this research, we try to establish a benchmark of work efficiency for software maintenance. To establish the benchmark, factors affecting work efficiency should be clarified, using a dataset collected from various organizations (cross-company dataset). We used dataset includes 134 data points collected by Economic Research Association in 2012, and analyzed factors affected work efficiency of software maintenance. We defined the work efficiency as number of modified modules divided by working time. The main contribution of our research is illustrating factors affecting work efficiency, based on the analysis using cross-company dataset and working time. Also, we showed work efficiency, classified the factor. It can be used to benchmark an organization. We empirically illustrated that using Java and restriction of development tool affect to work efficiency.**

Keywords—cross-company dataset; linear regression; work efficiency; working time;

I. INTRODUCTION

Enterprise software needs software maintenance when a business process is changed. It often occurs, and hence users sometimes contract software maintenance with companies. Software maintenance does not mean only removing faults found after software release. Software needs extensions or modifications of its functions due to changes in a business environment, and software maintenance also indicates them. ISO/IEC 14764 [5] classifies software maintenance into followings:

- Corrective maintenance: modifications of faults found after software release.
- Preventive maintenance: corrective modifications before potential faults become actual faults, after software release.
- Adaptive maintenance: modifications to keep software availability against environmental changing after software release.
- Perfective maintenance: modifications for conservation or improvement of software performance or maintainability after software release.

It is important to establish a benchmark (reference values to compare an organization's work efficiency with others [8]) of work efficiency for software maintenance. For organizations which offer software maintenance service, the benchmarking is the basis of process improvement. The process improvement will enhance price competitiveness of the companies. For users (customer of software maintenance), the benchmarking is useful to evaluate work efficiency of the service supplier of software maintenance. If the work efficiency is low, the price of software maintenance may be higher than other service suppliers, and it gives the chance to reconsider the contract with the supplier.

In this research, we try to establish a benchmark of work efficiency for software maintenance. To establish the benchmark, factors affecting work efficiency (e.g., System architecture) are clarified first, and then the dataset is stratified by the factors, using a dataset collected from various organizations (cross-company dataset). When using the benchmark, one compares work efficiency with a reference value whose factor (e.g., System architecture) corresponds to the target. Note that the maintenance we focused on is only software maintenance, and it does not include system maintenance.

Based on a preliminary analysis, we regarded working time for software maintenance per year as maintenance cost, and regarded the number of modified modules per year as amount of maintenance work. Based on them, we defined work efficiency as the number of modified modules divided by the working time. Although the amount of maintenance work was also measured by Function Point Analysis method, it includes many missing values. So, we used the number of modified modules alternatively. We analyzed the relationships between working time and attributes such as programming language and business sector, using multiple regression analysis. The attributes which increase working time depress work efficiency. In the analysis, we also show distributions of work efficiency for each attribute, using boxplots.

The main contribution of our research is illustrating factors affecting work efficiency, based on the analysis using cross-company dataset and working time recorded on it. Also, we showed work efficiency, classified the factor. It can be used to benchmark an organization. We empirically illustrated that using Java and restriction of development tool affect to work efficiency. Note that we considered a kind of maintenance in the analysis, using a variable of maintenance type. The detail of the variable is explained in the section 2.

II. DATASET

The dataset used in the analysis includes 134 data points of the software maintenance agreement (project) which were collected from 120 organizations in 2012 by the Economic Research Association. They send questionnaires to companies, and based on the responses, the dataset was made. Hence, we did not know how to record each attribute in detail. Note that generally, cross-company dataset is collected by similar way (e.g., Cross-company dataset [4] collected by ISBSG [3]). 107 data points are business software, and 74 data points are fixed price contract (Software maintenance is performed during certain periods with fixed price [11]). The data points were collected mainly from software maintenance service suppliers. The number of modified modules and working time were collected in a year.

Attributes analyzed in this research are described in Table I. On some data points, multiple programming languages were used. We picked up programming language whose usage rate was more than 50% of the data points. We assumed programming languages whose usage rate was less than 50% is not affected very much, because the combination of the languages was not very various. For example, Java and HTML are often used together, and in this case, focusing only Java is proper. System architecture was settled in the same way. Maintenance type is the classification based on the rate of maintenance activities. When the rate was more than 50%, the value of the attribute was set as corrective, preventive, adaptive, or perfective.

In Table I, attributes from human factor to tool factor (We call them productivity factors) are defined based on [11], and they were evaluated on a three-point scale (Low value indicates a severe condition, i.e., Work efficiency may be decreased). They indicate the degree of difficulties of the factors. The number of analyzed data points was different on each attribute, because each attribute includes missing values. To handle missing values on regression models, we applied listwise deletion [7], which is widely used for statistical analysis.

TABLE I. DESCRIPTION OF ATTRIBUTES.

Attribute	Description
Maintenance cost	Cost of software maintenance per year (Price on each contract)
Working time of service supplier	Working time for software maintenance of the service supplier per year
Number of engineers of service supplier	The number of engineers of the service supplier (including full-time and part-time workers)
Total working time	Total working time for software maintenance of the user and the service supplier per year
Total number of engineers	Total number of engineers of the users and the service supplier (including full-time and part-time workers)
Work efficiency	The number of modified modules / total working time
FP of modified spots	Amount of modified functions per year measured by Function Point Analysis method.
SLOC of modified spots	Amount of modified source lines of code per year
Number of modified modules	Number of modified modules per year
Number of modified screens	Number of modified screens per year
Number of modified reports	Number of modified reports per year
Number of modified data files	Number of modified data files per year
Number of modified batch files	Number of modified batch files per year
Maintenance type	Corrective, preventive, adaptive, and perfective
System architecture	Mainframe, Web system, and Client-Server
Programming language	SQL, Java, Java Script, Visual Basic, COBOL, HTML, C, JSP, C++ et al.
Business sector	Manufacturing, Wholesale & retail, banking & insurance, Service industry, Electronics & computers et al.
Human factor	Difficulties about size of project (or organization) and level of skill
Problem factor	Difficulties about type, importance, relationships, restriction, and ramification of problems
Process factor	Difficulties about programming language and software development methodology
Product factor	Difficulties about reliability, size, control structures, and complexity of the software
Resource factor	Difficulties about hardware, duration, and budget
Tool factor	Difficulties about library, complier, test tool, maintenance tool, and reverse engineering tool

There is various kinds of software. It is considered in the analysis, because system architecture, programming language and business sector are used as explanatory variables, and they are considered to denote the kinds of software indirectly. For example, when business sector is banking and programming language is COBOL, the software is considered as main software of banking system.

TABLE II. DESCRIPTION OF ATTRIBUTES CORRELATION COEFFICIENTS BETWEEN MAINTENANCE COST AND WORKLOAD.

Attribute	Number of data point	Correlation coefficient	p-value
Number of engineers of service supplier	76	0.60	0.00
Working time of service supplier	81	0.79	0.00

TABLE III. THE MODEL USING THE NUMBER OF THE SERVICE SUPPLIER'S STAFF AND WORKING TIME OF THE THE SERVICE SUPPLIER.

Attribute	Standardized partial regression coefficients	p-value	VIF
Working time of service supplier	0.76	0.00	1.6
Number of engineers of service supplier	0.15	0.07	1.6

We defined a new attribute, work efficiency. It is the ratio of output to input human resources (see Table I). We treated the number of modified modules as the output, and working time as the input. The definition is based on the preliminary analysis described in section III.

III. PRELIMINARY ANALYSIS

We used Spearman's rank correlation coefficient to avoid influence of outliers. In what follows, "correlation" indicates the Spearman correlation. We applied the multiple linear regression analysis, and analyzed the relationships between independent variables and a dependent variable. It can handle relationships between independent variables (i.e., confounding). Mainly, using the multiple linear regression analysis has two advantages. One is we can compare the explanatory power between models. The other is we can eliminate confounding between independent variables.

As a rule of thumb, when adjusted R^2 of built model is larger than 0.50, the model has adequate explanatory power toward the dependent variable [2]. The threshold is reasonable, especially when analyzing software project dataset. Since the project dataset affect human factor to some extent, but the factor is not a dominant factor. In the regression analysis, ratio scale attributes were log transformed, to avoid influence of outliers. We set the significance level at 0.05.

A. Attributes Related to Maintenance Cost

As the preliminary analysis, we analyzed attributes related to maintenance cost. Software maintenance cost is mainly based on the labor cost. So, it is considered to be mainly settled based on working time or the number of engineers of the service supplier. To see the effect of the attributes to maintenance cost, we calculated correlations of them. Table II shows the correlations to the maintenance cost. The working time had a stronger correlation than the number of engineers.

TABLE IV. CORRELATION COEFFICIENTS BETWEEN FP MODIFIED AND OTHER AMOUNT OF MODIFICATIONS.

Attribute	Number of data point	Correlation coefficient	p-value
Number of modified modules	14	0.54	0.04
SLOC of modified spots	22	0.37	0.09
Number of modified screens	17	0.28	0.28
Number of modified reports	13	0.23	0.45
Number of modified data files	14	0.46	0.10
Number of modified batch files	14	-0.06	0.84

TABLE V. COEFFICIENTS BETWEEN WHOLE WORKING TIME AND AMOUNT OF MODIFICATIONS.

Attribute	Number of data point	Correlation coefficient	p-value
Number of modified modules	46	0.47	0.00
FP of modified spots	14	0.65	0.01

TABLE VI. COEFFICIENTS FOR WHOLE STAFF AND AMOUNT OF MODIFICATIONS.

Attribute	Number of data point	Correlation coefficient	p-value
Number of modified modules	45	0.58	0.00
FP of modified spots	14	0.53	0.05

The correlation between the working time and the number of engineers is 0.56. So, they may affect each other. To handle the mutual relationship (i.e., confounding), we applied multiple linear regression analysis, treated maintenance cost as dependent variable. Table III shows standardized partial regression coefficients of the model. The working time has larger coefficient and its p-value is smaller than 0.05. Adjusted R^2 of the model is 0.73. So, it has explanatory power toward maintenance cost to some extent.

On the model, the values of the variance inflation factor (VIF) of the variables were smaller than 10, and the conditional index was smaller than 30. The value was 8.3. It means there is not multicollinearity on the model [12]. Note that although the number of modified modules affects working time and a number of engineers, it does not affect maintenance cost directly. So, we did not include it as an independent variable.

The results mean maintenance cost is mainly settled based on working time of the service supplier. So, decreasing working time lessens maintenance cost. That is, analyses of working time and work efficiency are regarded as analysis based on maintenance cost. Although the results are not surprising, it is necessary to enhance reliability of the analysis. To our knowledge, there is no research which analyzed relationships between maintenance cost, working hour and the number of engineers.

TABLE VII. THE MODEL USING THE SYSTEM ARCHITECTURE.

Attribute	Standardized partial regression coefficients	p-value	VIF
Number of modified modules	0.51	0.00	1.0
Web system	0.35	0.00	1.0

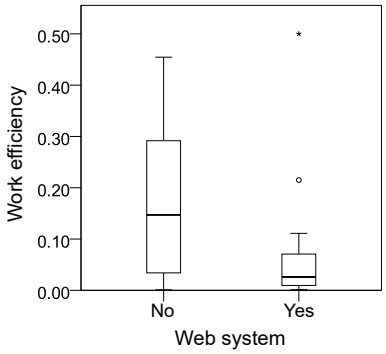

Fig. 1 Work efficiency of the web based system.

TABLE VIII. THE MODEL USING THE PROGRAMMING LANGUAGE.

Attribute	Standardized partial regression coefficients	p-value	VIF
Number of modified modules	0.58	0.00	1.0
Java	0.44	0.00	1.0

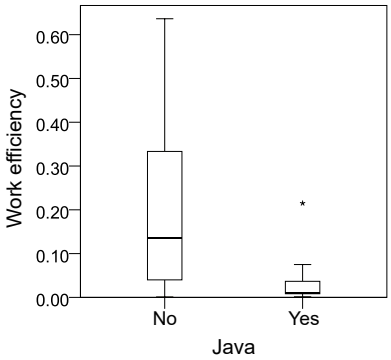

Fig. 2 Work efficiency of Java language.

B. *Attribute Indicating Amount of Modification*

To define work efficiency of software maintenance, an attribute indicating the amount of modification is needed. Although FP of modified spots is most appropriate to the amount, it has many missing values. So, we identified the attribute which has lesser missing values, and has a strong relationship to FP of modified spots. It is used as the attribute indicating amount of modification in the subsequent analyses.

Candidates of the attribute are SLOC of modified spots, the number of modified modules, the number of modified screens, the number of modified reports, the number of modified data files, and the number of modified batch files. The correlations between FP of modified spots and them are shown in Table IV. The number of modified modules has strongest correlation and its p-value was smaller than 0.05. So, we regarded it as the attribute indicating amount of modification.

Note that the correlation between the number of modified modules and FP of modified spots is 0.54. It is not very strong relationship, and therefore analysis results based on the number of modified modules may be different from the results based on the FP of modified spots.

In the following analysis, we did not care software size (i.e., Total number of modules). This is because our previous study [13] showed it does not affect a number of engineers.

C. *Relationship between Work Amount and Amount of Modification*

We analyzed the relationship between the work amount and the amount of modification. This is a preliminary analysis of work efficiency. In the analysis, we used total work amount of the user and the service supplier. This is because although maintenance cost is affected by the work amount of the service supplier only, amount of modification is affected by total work amount of the user and the service supplier. Note that activities of software maintenance are performed not only the service supplier but also the user.

We analyzed the relationship between the number of modified modules and total working time of the user and the service supplier. Additionally, we analyzed the relationship between FP of modified spots and the total number of engineers of the user and the service supplier. The analysis was performed to validate using the number of modified modules and the total working time, to analyze work efficiency.

The correlations between work amount and the amount of modification are shown in Table V and VI. The strength of the relationship between the number of modified modules and the total working time is moderate. Similarly, the strength of the relationships between other attributes of work amount and modification is also moderate. That is, the strength of all relationships is almost same, and it means analysis based on the number of modified modules and the total working time is not inappropriate.

We analyzed the relationship between the number of modified modules and the total working time, using simple linear regression analysis. This is because R^2 of the model can be used as the reference value of the subsequent analyses. R^2 of the model is 0.32. This means the number of modified modules is not enough to settle the total working time.

IV. ANALYSIS OF WORKING TIME

In this section, we applied multiple linear regression analysis. On the models, we treated the total working time of the user and the service supplier as the dependent variable. Using the multiple linear regression analysis, we can consider confounding factors. That is, the standardized partial regression coefficients is calculated by eliminating the influence of confounding.

TABLE IX. THE MODEL USING THE BUSINESS SECTOR.

Attribute	Standardized partial regression coefficients	p-value	VIF
Number of modified modules	0.58	0.00	1.0
Banking and insurance	0.24	0.05	1.0

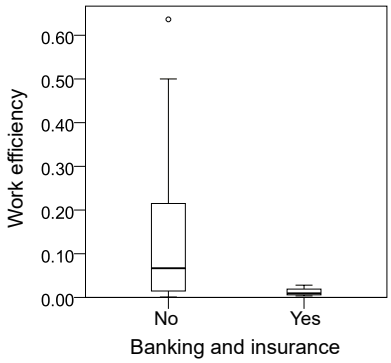

Fig. 3 Work efficiency of banking and insurance.

TABLE X. THE MODEL USING THE PRODUCTIVITY FACTORS.

Attribute	Standardized partial regression coefficients	p-value	VIF
Number of modified modules	0.44	0.00	1.1
Tool factor	-0.39	0.00	1.1

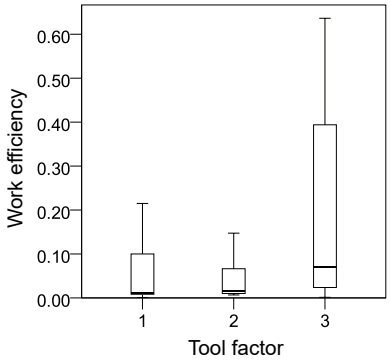

Fig. 4 Work efficiency of tool factor.

In the models, we used the number of modified modules as an independent variable. This means we can consider the influence of the number of modified modules when building the regression model. Note that we did not care software size (i.e., Total number of modules). This is because it did not affect a number of engineers in our previous study [13].

We also showed relationships between work efficiency and attributes using boxplots. Concretely speaking, based on standardized partial regression coefficients of the models, attributes affecting the total working time were identified, and distributions of work efficiency stratified by the attributes were shown using boxplots. The boxplot is used for the benchmarking.

In the boxplots, the bold line in each box indicates the median value. Small circles indicate outliers, that is, values that are more than 1.5 times larger than the 25%-75% range from the top of the box edge. Stars indicate extreme outliers, whose values are more than 3.0 times larger than this range. Some outliers do not include the boxplots to improve readability of them.

Before applying multiple linear regression analysis, we have made dummy variables to handle nominal scale attributes. When building the models, we applied stepwise variable selection. Variables were included when the p-value is smaller than 0.05, and excluded when the p-value is larger than 0.1.

A. Effect of System Architecture

We analyzed the relationship between system architecture and work amount, i.e., The total working time. On the multiple linear regression analysis, the number of modified modules and system architecture were treated as the independent variables, and the total working time was treated as the dependent variable. As a result, adjusted R^2 of the model is 0.43. Although adding system architecture as the independent variable improved adjusted R^2 from 0.32 (see section 3.C) to 0.43, the value is smaller 0.50. So, the independent variables are not sufficient to explain the total working time.

Standardized partial regression coefficients of the model are shown in Table VII. On the model, the values of the VIF of the variables were smaller than 10, and the conditional index was 4.9. So, there is not multicollinearity on the model. Variable selection deleted dummy variables of system architecture except for the variable of the Web system. The standardized partial regression coefficients had positive value. This means software on Web architecture has a larger working time than others, when the number of modified modules (i.e., Amount of modification) of them is same.

To benchmark software maintenance activities, we show a boxplot of the Web system and work efficiency in Fig. 1. The boxplot illustrates software on Web system has a lower work efficiency. This may be because the web system is sometimes not structured, and it makes lower work efficiency of maintenance.

B. Effect of Programming Language

We focused on the effect of programming language to the total working time. We treated the number of modified modules and programming language as the independent variables, and the total working time as the dependent variable in the multiple linear regression analysis. On the built model, adjusted R^2 is 0.52. So, using both the number of modified modules and programming language is effective, if users or service suppliers try to settle the total working time using a multiple linear regression model.

Table VIII shows standardized partial regression coefficients of the model. Variable selection chose the dummy variable of the Java language, and eliminated other variables of programing language. On the model, the values of the VIF of

TABLE XI. THE MODEL USING THE MAINTENANCE TYPE.

Attribute	Standardized partial regression coefficients	p-value	VIF
Number of modified modules	0.60	0.00	1.1
Corrective	-0.13	0.38	1.6
Adaptive	-0.27	0.07	1.5

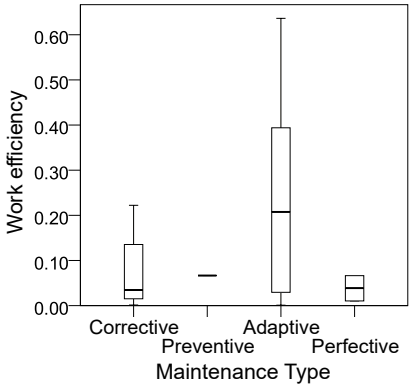

Fig. 5 Work efficiency of maintenance type.

TABLE XII. THE MODEL USING THE MULTIPLE FACTORS.

Attribute	Standardized partial regression coefficients	p-value	VIF
Number of modified modules	0.47	0.00	1.1
Java	0.37	0.00	1.1
Tool Factor	-0.29	0.01	1.2

the variables were smaller than 10, and the conditional index was 4.7. This means there is no multicollinearity on the model.

The standardized partial regression coefficients had positive value. Therefore, the total working time is increased when used programming language is Java on the maintained software. The correlation between Java and the web system was 0.46. The result suggests Java is often used with the web system, and work efficiency of Java may be affected by the web system.

For benchmarking of software maintenance, relationships between Java and work efficiency is shown by the boxplot in Fig. 2. On the figure, distribution of work efficiency depends on whether Java is used or not.

C. *Effects of Business Sector*

We analyzed the relationship between the total working time and the business sector where software is used. The effect of business sector was analyzed using a multiple linear regression model. On the model, adjusted R^2 is 0.37. The number of modified modules and business sectors are not sufficient to settle the total working time, since adjusted R^2 is smaller than 0.50.

Standardized partial regression coefficients of the model are shown in Table IX. On the model, the values of the VIF of the variables were smaller than 10, and the conditional index was 4.4. So, there is not multicollinearity on the model.

Variable selection deleted dummy variables of business sector except for the variable of banking and insurance. The standardized partial regression coefficients had positive value. This suggests working time becomes larger when software for banking and insurance is maintained. This may because the banking and insurance system needs high reliability, and it makes work efficiency lower.

Fig. 3 is the boxplot which shows the distribution of work efficiency based on whether business sector is banking and insurance or not. When business sector is banking and insurance, work efficiency is explicitly lower.

D. *Effects of Productivity Factors*

We analyzed on the effect of productivity factors to the total working time, using multiple linear regression analysis. On the built model, adjusted R^2 is 0.41. The value is smaller than 0.50, and therefore using the number of modified modules and productivity factors is not enough to settle the total working time.

Table X shows standardized partial regression coefficients of the model. On the model, the values of the VIF of the variables were smaller than 10, and the conditional index was 10.0. That is, there is not multicollinearity on the model.

Variable selection chose the tool factor, and eliminated other variables of productivity factors. The standardized partial regression coefficients had negative value. Therefore, the total working time is decreased when the value of the tool factor is larger (i.e., The demand for the factor is not severe). When tool factor (e.g., maintenance tools) can be changed, it should be done to enhance work efficiency.

The relationships between the tool factor and work efficiency is shown by the boxplot in Fig. 4. Although we do not show boxplots of other productivity factors, work efficiency is higher when the values of the factors are 3. Especially, the difference of work efficiency is explicit on the boxplot of the tool factor.

E. *Effects of Maintenance type*

We analyzed the relationship between maintenance type and the total working time. Although the effect of maintenance type was analyzed using a multiple linear regression model, variable selection deleted all dummy variables of maintenance type. So, we used dummy variables of corrective maintenance and adaptive maintenance to build the model without variable selection. On the model, adjusted R^2 is 0.41. The number of modified modules and maintenance type are not sufficient to settle the total working time, since adjusted R^2 is smaller than 0.50.

Standardized partial regression coefficients of the model are shown in Table XI. On the model, the values of the VIF of the variables were smaller than 10, and the conditional index was 6.7. So, there is not multicollinearity on the model.

On the dummy variable of adaptive maintenance, p-value is close to 0.05, and the standardized partial regression coefficients had a negative value. Adaptive maintenance may diminish working time. Note that the observation is not statistically significant, and therefore more analysis is needed to determine the effect of maintenance type.

Fig. 5 is the boxplot which shows the distribution of work efficiency based on maintenance type. When the maintenance type is adaptive, work efficiency is rather higher than corrective.

F. *Effects of Multiple Attributes*

In section IV. A to E, we assumed that users or service suppliers benchmark their activity, focusing on a single attribute. To support it, we analyzed the relationships between the total working time and each attribute. For example, to support benchmarking based on system architecture, we showed the effect of system architecture to the total working time, using multiple linear regression analysis. Additionally, we showed distributions of work efficiency based on the system architecture using a boxplot.

This section showed dominant attributes which affect the total working time. The candidates are attributes which we picked up in section IV. A to E. To perform it, we applied multiple linear regression analysis. Candidates of independent variables are web system (architecture), Java (programming language), tool factor (productivity factors), banking and insurance (business sector), adaptive (maintenance type), and the number of modified modules.

Table XII shows standardized partial regression coefficients of the model. On the model, the values of the VIF of the variables were smaller than 10, and the conditional index was 11.4. Hence, there is not multicollinearity on the model.

Variable selection chose the dummy variable of Java, tool factor, and the number of modified modules. It eliminated the dummy variables of the web system, adaptive maintenance, and banking and insurance. So, they do not affect the total working time when multiple attributes are considered. The standardized partial regression coefficients of tool factor had a negative value. Therefore, the total working time is decreased if the demand for the tool factor is not severe, even when multiple attributes are considered.

On the built model, adjusted R^2 is 0.56. The value is larger than 0.50, and therefore using the tool factor, the number of modified modules, and the dummy variable of Java is enough to settle the total working time.

Comparing absolute value of the regression coefficients, the effect of the number of modified modules to the total working time is the largest, and whether using Java or not affects it moderately. The effect of tool factor is the smallest. However, p-value of the regression coefficient is smaller than 0.05, and therefore tool factor also affects the total working time.

V. Related Work

Some researches analyzed work efficiency factors of software maintenance. Jørgensen [6] analyzed software company dataset, and showed that work efficiency is not affected by programming language. Ahn et al. [1] used variables which are similar to the productivity factors in a software maintenance effort estimation model. However, these researches did not analyze cross-company dataset.

ISBSG (International Software Benchmarking Standards Group) collects cross-company dataset of software maintenance [4]. Tsunoda et al. [14] analyzed the dataset, and they concluded that only several companies data can be used to analyze work efficiency on the dataset, due to missing values. In contrast, we analyzed data points collected from many companies (roughly speaking, each data point was collected from each company). So, the analysis results of this research are expected to have high generality.

There are few reports or researches which analyzed cross-company software maintenance dataset. Japan Users Association of Information Systems (JUAS) and Ministry of Economy, Trade and Industry used the cross-company dataset, and showed work efficiency stratified by business sector [9]. They defined maintenance cases per engineer as work efficiency, and their definition is a bit rough, compared with our definition. Also, JUAS reports the following results about software maintenance [10].

- Some rate of working time is used for communications about stakeholders.
- Requirement changes about software maintenance cause the delay of delivery time.
- Many companies conduct simultaneous modifications for reducing workload of software maintenance.
- Many companies adopt engineer awards programs for generating motivations.

We should consider the relationships between work efficiency and the factors in the future research. Note that the results are based on the summery of the questionnaire. That is, they did not analyze the relationships between work efficiency and the factors quantitatively in the report [10].

We analyzed factors related to work efficiency on software maintenance, using cross-company dataset [13]. In the research, work efficiency was defined as the number of modified modules per engineer. They defined work efficiency based on the number of engineers, and therefore their definitions are rather rough. In contrast, this research defines work efficiency based on working time. Although it is not easy for users to grasp working time for software maintenance, the definition is more precise.

VI. Conclusions

In this research, we tried to establish a benchmark of work efficiency for software maintenance. The main contribution of the research is that we used work efficiency based on working time, and the dataset collected from many companies, to clarify the attributes related to the work efficiency. We think it is possible for users and service suppliers to benchmark their

activities using boxplots shown in this research. Note that the benchmarking should be used as reference, but not as rigid criteria. Since we did not use FP of modified spots, and the variance of work efficiency is large, and therefore more analysis is needed.

ACKNOWLEDGMENT

This research was partially supported by the Japan Ministry of Education, Science, Sports, and Culture [Grant-in-Aid for Scientific Research (C) (No. 25330090)]

REFERENCES

[1] Y. Ahn, J. Suh, S. Kim, and H. Kim, "The software maintenance project effort estimation model based on function points," Journal of Software Maintenance: Research and Practice, Vol. 15, No. 2, pp. 71-85, 2003.

[2] R. Brook, and G. Arnold, Applied Regression Analysis and Experimental Design, CRC Press, 1985.

[3] International Software Benchmarking Standards Group, https://www.isbsg.org/

[4] International Software Benchmarking Standards Group (ISBSG), ISBSG Maintenance & Support Data Suite Release 4, ISBSG, 2010.

[5] ISO/IEC 14764:2006, Software Engineering - Software Life Cycle Processes - Maintenance, International Organization for Standardization, 2006.

[6] M. Jørgensen, "Experience With the Accuracy of Software Maintenance Task Effort Prediction Models," IEEE Transactions on Software Engineering, Vol. 21, No. 8, pp. 674-681, 1995.

[7] R. Little, and D. Rubin, Statistical Analysis with Missing Data, 2nd ed., , John Wiley & Sons, 2002.

[8] C. Lokan, T. Wright, P. Hill, and M. Stringer, "Organizational Benchmarking Using the ISBSG Data Repository," IEEE Software, Vol. 18, No. 5, pp. 26-32, 2001.

[9] Japan Users Association of Information Systems (JUAS), Software Metric Survey, JUAS, 2008 (In Japanese).

[10] Japan Users Association of Information Systems (JUAS), Software Metric Survey 2013, JUAS, 2013 (In Japanese).

[11] Software Evolution Research Consortium (SERC), Software Evolution Research Consortium Report, SERC, 1995.

[12] B. Tabachnick, and L. Fidell, Using Multivariate Statistics, 3rd ed., Harper Collins College Publishers, 1996.

[13] M. Tsunoda, A. Monden, K. Matsumoto, and T. Oshino, "Analysis of Software Maintenance Efficiency Focused on Process Standardization," Proc. of International Workshop on Empirical Software Engineering in Practice (IWESEP), pp.3-8, Nara, Japan, 2011.

[14] M. Tsunoda, and K. Ono, "Pitfalls of Analyzing a Cross-company Dataset of Software Maintenance and Support," Proc. of International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), pp.349-354, Las Vegas, NV, USA, 2014.