Analysis of Software Maintenance Efficiency Focused on Process Standardization

Masateru Tsunoda
Nara Institute of Science and Technology
Kansai Science City, 630-0192
Japan
masate-t@is.naist.jp

Akito Monden
Nara Institute of Science and Technology
Kansai Science City, 630-0192
Japan
akito-m@is.naist.jp

Ken-ichi Matsumoto
Nara Institute of Science and Technology
Kansai Science City, 630-0192
Japan
matumoto@is.naist.jp

Tomoki Oshino
Economic Research Institute,
Economic Research Association
Higashi-Ginza Mitsui Bldg. 5-13-16,
Ginza, Chuo-ku, Tokyo, 104-0061 Japan
er352@zai-keicho.or.jp

ABSTRACT
In this research, to establish a benchmark for software maintenance efficiency, we analyzed factors affecting the work efficiency and showed reference value stratified the factors. We analyzed dataset of software maintenance collected from 83 organizations. Attributes recorded in the dataset are standardization status of organization, system architecture, the number of engineers, the number of base modules, the number of modified modules, and so on. In the analysis, we regarded modified modules per engineer as a work efficiency index, and clarified relationships to other attributes. As a result, we identified that standardization status of organization is important for work efficiency.

Categories and Subject Descriptors
K.6.3 [**Management of Computing and Information Systems**]: Software Management – *Software maintenance*, **Management of Computing and Information Systems**]: general – *Economics*.

General Terms
Management, Measurement, Economics.

Keywords
Module modification, productivity, cross-company dataset, benchmarking.

1. INTRODUCTION
Recently, a number of software users (companies or organizations) contract with software developers (companies) for maintenance of enterprise software, and therefore the agreement of software maintenance becomes more important. 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 classifies software maintenance into followings:

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

In this research, we try to establish a benchmark (reference values to compare an organization's work efficiency with others [5]) of work efficiency for software maintenance contract. 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 dataset collected from various organizations (cross-company dataset). When using the benchmark, compare work efficiency with a reference value whose factor (e.g. system architecture) is correspond to the target. We focus on the standardization of software maintenance process in the analysis. When standardization status (process is standardized or not) has strong relationships to work efficiency, reference values stratified standardization status can be used to confirm difference of work efficiency between standardized organizations and not standardized ones. That is useful to decide whether software maintenance process should be standardized or not. Major contribution of our research is to clarify relationships between standardization status and maintenance efficiency using cross-company dataset.

The dataset used in the analysis was collected from 83 organizations in 2007 by Economic Research Association [2]. We analyzed relationships between work efficiency and other attributes such as system architecture, in addition to standardization status. The dataset does not have enough cases to use analysis results as rigorous benchmark. So, values shown in the research should be

Table 1. Description of dataset

Attribute	Description
Process standardization	Status of standardization of software maintenance process (process is standardized, standardization is work in progress, or maintenance process is not standardized)
System architecture	System architecture on which the software runs (client–server system, Web based system, mainframe system)
Number of base modules	Total number of modules included in the software
Number of modified modules	The number of modified modules in the maintenance activity in a year
Number of engineers	The number of contractor's resident engineers
Modified modules per engineer	The number of modified modules / number of engineers
Modified modules per engineer and base	The number of modified modules / the number of engineers / the number of base modules
Human factor	Degree of difficulties about size of project (or organization) and level of skill
Problem factor	Degree of difficulties about type, importance, relationships, restriction, and ramification of problems
Process factor	Degree of difficulties about programming language and software development methodology
Product factor	Degree of difficulties about reliability, size, control structures, and complexity of the software
Resource factor	Degree of difficulties about hardware, duration, and budget
Tool factor	Degree of difficulties about library, complier, test tool, maintenance tool, and reverse engineering tool

used casually. Nevertheless, we think our result is effective because there are very few researches or reports which analyzed software maintenance efficiency using cross-company dataset.

In the analysis, we regarded modified modules per engineer as a work efficiency index. That is, the number of engineers was considered as inputs, and the number of modified modules in a year was considered as outputs (High modified modules per engineer indicates high work efficiency). It is more precise that a work efficiency index is defied using software maintenance effort and modified lines of code (or function point). However, modified lines of code (and function point) in the dataset used includes many missing values, and software maintenance effort is not recorded. In addition, it is easier for software uses (companies or organizations) to measure the number of modified modules than modified lines of code or function point (Actually, the dataset is almost collected from software uses). Similarly, it is easier to measure the number of engineers than software maintenance effort. So using the number of modified modules and the number of engineers makes it easy for software users to refer the benchmark.

In what follows, Section 2 explains dataset used in the analysis. Section 3 shows analysis results. Section 4 introduces related works, and Section 5 concludes the paper with a summary.

2. DATASET

The dataset used in the analysis includes 83 cases of software maintenance agreement which were collected from 83 organizations in 2007 by Economic Research Association [2]. 78 cases are business software, and rest cases are factory automation software and other software. 46 cases are fixed price contract (Software maintenance is performed during certain period by fixed price [6]). The cases were collected mainly from software uses (companies or organizations). Each case is representative software maintenance agreement in each organization (Each organization provided one case), and number of modified module was collected in a year. The number of analyzed cases was different for attributes, because each attribute includes missing values. Attributes analyzed in this research are described in Table 1, and the followings are detailed explanations for some attributes.

- Process standardization status does not represent the status of a case included in the dataset, but the status of the entire organization which offered the case (Values of other attributes were collected in each software maintenance agreement).
- When process is standardized, standard process of software maintenance (sequence of activities such as analyzing, reviewing, documenting, and approval) is explicitly defined.
- Some cases have multiple system architectures (for example, a case of system architecture includes both mainframe and web based system).
- Modified modules per engineer and base is explained in section 3.2.
- Attributes from human factor to tool factor (We call them productivity factors) are defined based on [6], and they were evaluated on a three-point scale (Low value indicates severe condition, i.e., productivity may be decreased).

3. ANALYSIS RESULTS

3.1 Analysis Procedure

In the analysis, we analyzed influences of system architecture, productivity factor, the number of base modules, and the number of engineers to work efficiency (modified modules per engineer), in addition to influence of process standardization. When an attributes has considerable influence, the dataset was stratified by the attribute, to eliminate the influence of the attribute. For example, if system architecture had influence to work efficiency, the dataset was stratified by it, and a relationship between process standardization and work efficiency was analyzed. The dataset has missing values, so the number of analyzed cases is different in each analysis.

Table 2. Relationships between size attributes

Attribute		Number of base modules	Number of modified modules	Number of engineers	Modified modules per engineer
Number of base modules	ρ	1.00	0.70	0.23	0.60
	p-value		0.00	0.30	0.02
	Number of cases	35	25	22	15
Number of modified modules	ρ	0.70	1.00	0.56	0.83
	p-value	0.00		0.03	0.00
	Number of cases	25	25	15	15
Number of engineers	ρ	0.23	0.56	1.00	0.03
	p-value	0.30	0.03		0.92
	Number of cases	22	15	35	15
Modified modules per engineer	ρ	0.60	0.83	0.03	1.00
	p-value	0.02	0.00	0.92	
	Number of cases	15	15	15	15

When analyzing a relationship of ratio (or ordinal) scale attributes, we used Spearman's rank correlation coefficient to avoid influence of outliers. In what follows, "correlation" and ρ indicate the Spearman correlation. We did not apply a multivariate regression model because there are many missing values in the dataset. Instead of applying it, we stratified the dataset as mentioned above.

We used a box plot to analyze relationships between nominal scale attribute and ratio scale attribute. In a box plot, 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 figures to improve readability of them. Mann–Whitney U test was applied to confirm median is statistically different or not between two attributes. We set significance level as 5%.

3.2 Relationships between Size Attributes

We analyzed relationships between modified modules per engineer, the number of base modules, and the number of engineers, to confirm whether modified modules per engineer can be used as work efficiency index or not. It may be helpful in comprehending the analysis to regard the number of base modules as function point, the number of engineers as development effort, and modified modules per engineer as productivity.

Table 2 shows correlations of the relationships, and the major relationships are illustrated in Figure 1. Observations are as follows:

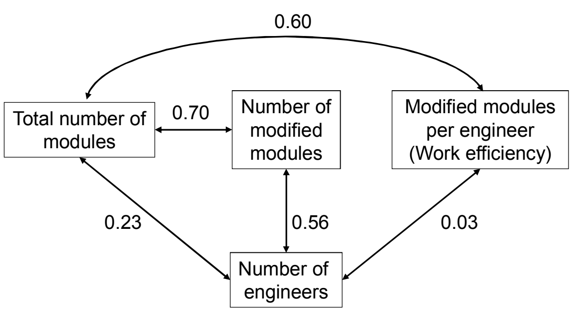

Figure 1. Major correlations between the number of base modules, number of engineers, and other attributes

- There is a positive correlation between the number of modified modules and the number of engineers (ρ = 0.56). So, when the number of engineers is large, the number of modified modules is also large.
- The correlation between the number of engineers and the number of base modules is not large (ρ = 0.23). Therefore, size of entire software does not greatly affect the number of engineers.
- The correlation between the number of engineers and modified modules per engineer is very small (ρ = 0.03). This does not mean fewer engineer increases modified modules per engineer.

The number of base modules has significant correlation to modified modules per engineer (ρ = 0.60) and the number of modified modules (ρ = 0.70). It may be likely that when the number of base modules is large, the number of modified modules is also large, and as a result, modified modules per engineer becomes large. So we defined modified modules per engineer and base (the number of modified modules / the number of engineers / the number of base modules), to eliminate influence of the number of base module, and used it to strengthen analysis results.

3.3 Relationship to System Architecture

Figure 2 and Figure 3 show boxplots of modified modules per engineer and modified modules per engineer and base for three different system architectures. In modified modules per engineer (Figure 2), mainframe system is the highest, and Web based system is the lowest. The median value of Web based system is 2.2 times smaller than client-server system, and 6.9 times smaller than mainframe system, as shown in Table 3. However, they are not statistically different. There are not enough cases in the dataset (see Table 3), and it may affect statistical test results.

Also, in modified modules per engineer and base, mainframe system is the highest while the distribution is wide, as shown in Figure 3. The distribution (position and size of the box in the figure) of Web based system is almost same as client-server system. The median value of Web based system is same as client-server system,

Table 3. Median values of modified modules per engineer (and base) stratified by system architecture

	System architecture	Number of cases	Median	p-value (difference from Web based)	p-value (difference from client-server)
Modified modules per engineer	Client-server	9	40.0	88%	-
	Web Based	11	18.0	-	88%
	Mainframe	8	125.0	21%	37%
	Total	28	55.0	-	-
Modified modules per engineer and base	Client-server	9	0.007	82%	-
	Web Based	11	0.007	-	82%
	Mainframe	8	0.011	27%	37%
	Total	28	0.008	-	-

Table 4. The number of base modules, the number of modified modules, and the number of engineers stratified by system architecture

System architecture		Number of base modules	Number of modified modules	Number of engineers
Client-server	Median	3000	125	10
	Number of cases	25	16	26
Web Based	Median	3000	180	10
	Number of cases	23	17	21
Mainframe	Median	4628	400	10
	Number of cases	21	14	20
Total	Median	3000	250	10
	Number of cases	69	47	67

and 1.6 times smaller than mainframe system (Table 3). Note that there is no statistically difference among them.

In the results, modified modules per engineer (and base) is larger when system architecture is mainframe system. Although we should carefully understand it because there is no statistically difference, it suggests system architecture should be considered when comparing work efficiency (modified modules per engineer).

We examined programming language used in each system. In the result, the most used programming language is Visual Basic in client-server systems, it is SQL in Web based systems, and it is COBOL in mainframe systems (Note that multiple programming languages are used in each system). Difference of work efficiency between system architectures may be affected by characteristics of programming languages.

We counted the number of base modules, the number of modified modules, and the number of engineers by system architecture (Table 4). Observations are as follows:

- Median values of the number of engineers are same among system architectures.
- In median values of the number of modified modules, mainframe system is 2.2 times larger than Web based system, and 3.2 times larger than client-server system.
- In median values of the number of base modules, mainframe system is 1.8 times larger than others.

In mainframe systems, it may be probable that the number of base

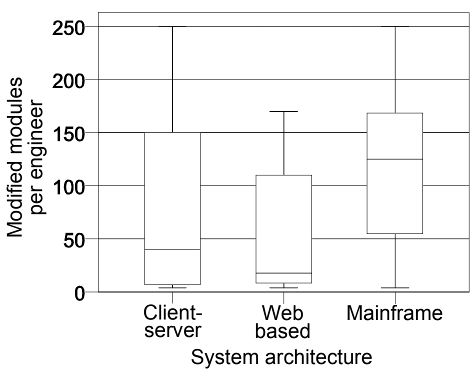

Figure 2. Modified modules per engineer stratified by system architecture

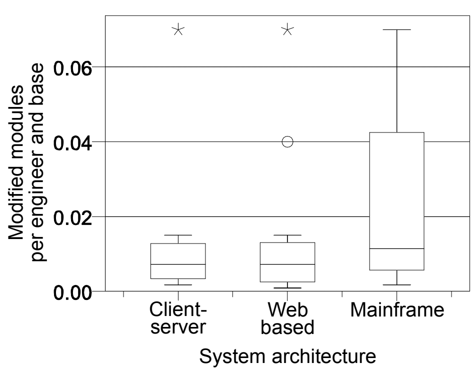

Figure 3. Modified modules per engineer and base stratified by system architecture

Table 5. Median values of modified modules per engineer (and base) stratified by process standardization

	Process standardization	Number of cases	Median	p-value (difference from standardized)	p-value (difference from not standardized)
Modified modules per engineer	Standardized	6	118.3	-	29%
	Work in progress	8	14.0	11%	89%
	Not standardized	1	7.1	29%	-
	Total	15	40.0	-	-
Modified modules per engineer and base	Standardized	6	0.026	-	57%
	Work in progress	8	0.005	2%	100%
	Not standardized	1	0.007	57%	-
	Total	15	0.008	-	-

Table 6. The number of base modules, the number of modified modules, and the number of engineers stratified by process standardization

Process standardization		Number of base modules	Number of modified modules	Number of engineers
Standardized	Median	3000	400	6
	Number of cases	13	10	11
Work in progress	Median	4000	100	10
	Number of cases	15	12	17
Not standardized	Median	1000	5	6
	Number of cases	7	3	7
Total	Median	3000	180	10
	Number of cases	35	25	35

modules increased the number of modified modules (see section 3.2). However, relative difference of the number of base modules between mainframe system and others is smaller than the number of modified modules. This reinforces the analysis result that work efficiency is different for system architecture.

3.4 Relationship to Process Standardization

Figure 4 illustrates a relationship between process standardization and modified modules per engineer, and Table 5 shows median values of them and statistical test results for their differences. In Figure 4, when process is standardized, modified modules per engineer is the highest, although the distribution is wide. The median value of “standardized” is 8.5 times larger than “work in progress,” as shown in Table 5. But they are not statistically different.

Same tendency was observed in modified modules per engineer and base. Figure 5 shows a relationship between process standardization and modified modules per engineer and base, and Table 5 presents median values of them and statistical test results. In Figure 5, the values of “standardized” are larger than “work in progress.” In Table 5, the median value of “standardized” is 5.2 times larger than “work in progress,” and there is statistically difference between them. The results indicate process standardization is related to work efficiency.

We checked the number of base modules, the number of modified modules, and the number of modules, stratifying the dataset according to process standardization. In Table 6, between “standardized” and “work in progress,” there is 0.6 times difference in the

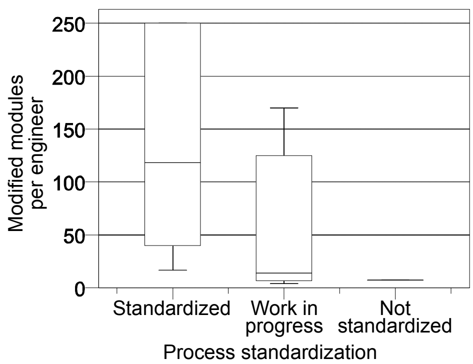

Figure 4. Modified modules per engineer stratified by process standardization

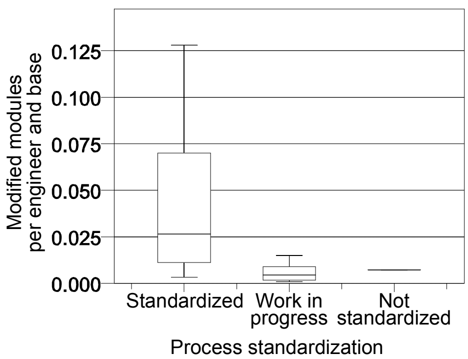

Figure 5. Modified modules per engineer and base stratified by process standardization

Table 7. Relationships to productivity factors

		Human factor	Problem factor	Process factor	Product factor	Resource factor	Tool factor
Modified modules per engineer	ρ	-0.09	0.38	0.20	0.32	-0.02	0.07
	p-value	76%	20%	51%	29%	96%	81%
	Number of cases	13	13	13	13	13	13
Modified modules per engineer and base	ρ	-0.26	-0.10	-0.23	-0.01	0.06	0.00
	p-value	39%	74%	46%	97%	85%	100%
	Number of cases	13	13	13	13	13	13

number of engineers, and 0.75 times difference in the number of base modules. Nevertheless, there is 4.0 times difference in the number of modified modules. So in this case, we do not have to care the positive correlation between the number of base modules and the number of modified modules (see section 3.2). This also supports the result that process standardization is related to work efficiency.

When stratifying the dataset by system architecture, both modified modules per engineer and modified modules per engineer and base are higher in process standardized organizations (except for modified modules per engineer in mainframe systems). Similarly, when stratifying the dataset by process standardization, they are higher in mainframe systems. Although we should be care that there are a few cases in each group after stratifying, both system architecture and process standardization are considered to affect work efficiency.

3.5 Relationship to Productivity Factors

Table 7 shows correlations between productivity factors and modified modules per engineer (and base). When a value of productivity factor is smaller, condition is more severe. That is, positive correlation indicates work efficiency decreases when the condition of the factor is severe.

Although the correlation between problem factor and modified modules per engineer is relatively higher (ρ = 0.38; not significant), the correlation to modified modules per engineer and base is low (ρ = -0.10). So, we cannot conclude problem factor affects work efficiency. Other factors have low correlation, and they are not statistically significant. In addition, their correlations are inconsistent between modified modules per engineer and modified modules per engineer and base. Namely, we did not observe explicit influences of productivity factors on work efficiency.

4. RELATED WORKS

Except for process standardization, some researches analyzed work efficiency factors on software maintenance. Jørgensen [4] analyzed software company dataset, and showed that work efficiency is not affected by the number of base modules and programming language. Ahn et al. [1] used variables which are similar to the productivity factors in a software maintenance effort estimation model. However, past researches did not clarified effect of process standardization. This is because they used dataset collected from a few companies, and it made the analysis of process standardization effect difficult.

There are very 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 (maintenance cases per engineer) stratified by business sector [3]. However they did not clarified the relationship between process standardization and work efficiency. Yokota [7] showed standardizing maintenance process is effective for work improvement, based on questionnaire data. But quantitative data analysis was not performed.

5. CONCLUSIONS

In this research, to establish a benchmark for software maintenance efficiency, we analyzed software maintenance data collected from 83 organizations. Modified modules per engineer is regarded as work efficiency index. We compared work efficiency by stratifying the dataset, and clarified process standardization status is related to work efficiency (modified modules per engineer). Our future work is collecting more data and analyzing it to enhance reliability of the results.

6. ACKNOWLEDGMENTS

This work is being conducted as a part of the StagE project, The Development of Next-Generation IT Infrastructure, and Grant-in-aid for Young Scientists (B), 22700034, 2010, supported by the Ministry of Education, Culture, Sports, Science and Technology.

7. REFERENCES

[1] Ahn, Y., Suh, J., Kim, S., and Kim, H. 2003. The software maintenance project effort estimation model based on function points. *Journal of Software Maintenance: Research and Practice*, 15, 2, 71-85.

[2] Economic Research Association. http://www.zai-keicho.or.jp/about/english.php

[3] Japan Users Association of Information Systems 2008. Software Metric Survey 2008. Japan Users Association of Information Systems.

[4] Jørgensen, M. 1995. Experience With the Accuracy of Software Maintenance Task Effort Prediction Models. *IEEE Transactions on Software Engineering*, 21, 8, 674-681.

[5] Lokan, C., Wright, T., Hill, P, and Stringer, M. 2001. Organizational Benchmarking Using the ISBSG Data Repository. *IEEE Software*, 18, 5, 26-32.

[6] Software Evolution Research Consortium 1995. Software Evolution Research Consortium Report, Software Evolution Research Consortium.

[7] Yokota, T. 2003. An Assessment for Software Maintenance Environment and its Improvement Examples. *Journal of the Society of Project Management*, 5, 2, 40-44.