Analyzing Effects of Cost Estimation Accuracy on Quality and Productivity

Osamu Mizuno, Tohru Kikuno
Graduate School of Engineering Science
Osaka University
1-3 Machikaneyama
Toyonaka-City, Osaka, 560, Japan
+81-6-850-6567
o-mizuno@ics.es.osaka-u.ac.jp

Katsumi Inagaki, Yasunari Takagi, Keishi Sakamoto
Social Systems Business Group
OMRON Corporation
Kusatsu-City, Shiga, 525, Japan
+81-77-565-5071
inagaki@eftses.krc.omron.co.jp

ABSTRACT

This paper discusses the effects of the estimation accuracy for software development cost on both the quality of the delivered codes and the productivity of the development team. The estimation accuracy is measured by metric RE(called relative error). Similarly, the quality and productivity are measured by metrics FQ(field quality) and TP(team productivity). Using actual project data on thirty-one projects at a certain company, the followings are verified by correlation analysis and test of statistical hypotheses: (1) There is a high correlation between the faithfulness of development plan to standards and the value of RE(A coefficient of correlation between them is -0.60). (2) Both FQ and TP are significantly different between projects with $-10\% < RE < +10\%$ and projects with $RE \geq +10\%$ (The level of significance is chosen as 0.05).

KEYWORDS

software project, development plan, software quality, productivity, test of statistical hypothesis

1 INTRODUCTION

In order to achieve high quality of software products and productivity of development team, various methodologies and technologies have been proposed. One of the most well-known methodologies is process improvement[6], and many theoretical investigations[3][11] as well as practical applications[7][12] have been reported. Some of them have already been actually implemented as tools[1].

This paper describes empirical research in a certain company, which we call Company A for convenience. Currently Company A is mainly developing systems with embedded software such as ATMs (Automated Teller Machine), POS (Point of Sales) terminals and ticket vending machines. In order to introduce a new methodology or tool into the field of development, it is necessary to show its effects and usefulness, and allow developers to understand it. This study is one such attempt.

In Company A, the process improvement activity has been conducted by the software engineering process group(SEPG). Especially, the following attempts have been carried out enthusiastically.

- Exhaustive collection of fundamental data[12][13].
- Establishing standards for activities.
  - Constructing the project plan.
  - Describing the development process[8][12].

As a result of these efforts, the following improvements have been observed in quality and productivity.

- The development plan document tends to be constructed faithfully to the standard of good writing.
- The development cost, which is one of the most important factors in the company, tends to be estimated accurately.
- Both the quality of the delivered code and the productivity of the development team tend to be stable and improving.

These improvements have led the development managers and the SEPG to believe the following proposition, which we call P_1: "If the cost estimation is accurate in the development plan, the quality of the product and the productivity of the team become stable and high." At present, verifying this proposition is one of the main problems for the manager and the SEPG.

On the other hand, the estimation of the quality and the productivity should be done for the project to be executed, and so it must be done at the beginning of the project. However, the accuracy of the cost estimation can only be calculated at the end of the project. That is, based on the proposition P_1, it is impossible to estimate the quality and the productivity at the beginning of the project (even if the proposition P_1 is proved to be correct).

Then, we try to prove the second proposition P_2: "If the plan is constructed faithfully to standards of good writing, then the cost estimation becomes accurate." In this paper, we aim to verify these two propositions using the data collected from actual development projects in Company A(unfortunately, the values of collected data cannot be published in this paper because of the conditions in the contract with Company A).

In this paper, we begin by discussing the proposition P_2. We evaluate and grade the development plan document by applying the procedure adopted for years in SEPG. The construction of the plan is evaluated with respect to the faithfulness to standards of good writing(the best grade is 50), and the development is evaluated with respect to the faithfulness to the plan(the best grade is 50).

Then, we calculate the cost estimation accuracy (called RE) defined by $RE = (actCOST - estCOST)/estCOST \times 100$ (%), of each project where $actCOST$ and $estCOST$ are the actual cost and estimated cost, respectively. Based on the calculated value of RE, we classify projects into three classes, C_0, C_+ and C_-: C_0 is a set of projects with $-10\% < RE < +10\%$, C_+ is a set of projects with $RE \geq +10\%$ and C_- is a set of projects with $RE \leq -10\%$. Here, we consider that ten percent is an important empirical threshold for the evaluation of RE.

In order to discuss the proposition P_2, we analyze the correlation between the grade of the development plan and the classification of projects based on the cost estimation accuracy RE. As a result, we show that there is a relatively high correlation between the faithfulness of development plan to standards and the value of RE(a coefficient of correlation between them is -0.60). The negative value of correlation coefficient means that the higher the faithfulness to standards becomes, the smaller the value of RE becomes(that is, the cost estimation becomes accurate).

Next, in order to discuss the proposition P_1, we analyze the following two relationships: (1) the relationship between the classification of projects based on RE and the quality of the delivered code, (2) the relationship between the classification of projects based on RE and the productivity of the team, by using the test of statistical hypothesis(t-test). As the result, we show that both the quality of the delivered code and the productivity of the team are significantly different between projects with $-10\% < RE < +10\%$ and projects with $RE \geq +10\%$ (The level of significance is chosen as 0.05).

2 BACKGROUND AND OBJECTIVE

2.1 Background

In Company A, process improvement has been undertaken to achieve high quality of software products and high productivity of the development team for the last five years[12]. The main improvement activities are summarized as follows:

(IM_1) Exhaustive but systematic collection of fundamental data from projects in Company A.

(IM_2) Establishing several key standards for developmental activities in projects.

The improvement activity IM_1 was executed according to software metrics recommendations[4][9][13], and the following D_1 – D_5 summarize the data to be used in this paper.

(D_1) the development plan document.

(D_2) the size of delivered code(measured by Kstep).

(D_3) the effort of each activity of the development process(measured by person-month).

(D_4) the number of faults detected and corrected during review, test and debug activities.

(D_5) the number of faults in the delivered code.

Among them, data D_1 is constructed before actual development starts, data D_2 – D_4 are taken after development is accomplished and code is delivered, and D_5 is collected during the six months after the code is delivered.

Next, the improvement activity IM_2 was focused on the construction of project plans and the formal definition of development process. After trial and error, Company A has succeeded in establishing standards of good writing for project plans. Additionally, software development process in Company A has been formally described using the generalized stochastic petri-net(GSPN), and a project simulator has already been implemented based on the GSPN[8].

As the result of these improvement activities over the past five years, the following changes C_1 – C_3 have been observed in Company A in the direction of improvement of quality and productivity.

(C_1) The development plan document (data D_1 in the improvement activity IM_1) tends to be constructed faithfully to standards of good writing(established in the improvement activity IM_2).

(C_2) The development cost, which is one of the most important factors for the company(but unfortunately is very difficult to estimate exactly[2][5]), tends to be estimated accurately.

(C_3) Both the quality of the delivered codes (data D_5 in the improvement activity IM_1) and the productivity of the development team (data D_2, D_3 in IM_1) tend to be stable and relatively better.

Both the development managers and the software engineering process group(SEPG) are eager to know the causes of the changes C_2 and C_3. To say more straightforwardly, they conjecture that the change C_1 is the cause of the change C_2 and that the change C_2 is the cause of the change C_3.

2.2 Objective

In this paper, we try to show the following three assertions A_1 – A_3 to answer the questions from the development managers and SEPG. The assertion A_1 is the same as the proposition P_2 in Section 1, and the assertions A_2 and A_3 correspond to the proposition P_1. For the analysis of A_1 – A_3, we apply correlation analysis and test of statistical hypothesis to actual data on thirty-one projects at Company A.

(A_1) In the project executed according to the project plan constructed faithfully to standards, the estimation error of the development cost is small.

The assertion A_1 implies that the faithfulness to standards in the construction of the project plan and the faithfulness to the project plan in the development are the main reasons for the small error in the cost estimation(that is, the difference between the estimated cost and the actual cost is small) of the project.

(A_2) In the project with the accurate cost estimation, the quality of the delivered code is high.

The assertion A_2 implies that the project, for which the cost estimation error is small, delivers the final software product with high quality. As mentioned in subsection 2.1, we evaluate the quality of the final software product using the data D_5 collected during the six months after its delivery.

(A_3) In the project with the accurate cost estimation, the productivity of the development team is high.

The assertion A_3 implies that in the project which is completed in accordance with estimate, the productivity of the development team is high (compared with the one in the so-called death march projects[15]).

3 DEFINITION OF METRICS

3.1 Process Model

In Company A, many kinds of computer control systems with embedded software are developed mainly using C language. The typical products are ATMs(Automated Teller Machine) for banking applications, POS(Point Of Sales) terminals for business applications and ticket vending machines for railroad applications. Such products are developed under the development process shown in Figure 1. The description of Figure 1 is one of the outcomes of the improvement activity IM_2 mentioned in subsection 2.1.

The process model shown in Figure 1 is a standard waterfall model. As will be described in detail in subsection 4.1, modification to the requirement specification is very rare and is limited only to layout of screens or the speed of CPU, and thus most of the requirement specification is decided by Company A. This may be one of the main reasons why the waterfall model shown in Figure 1 is still effectively and successfully used in the company. Strictly speaking, some kinds of irregular control flows (such as backwards flow to previous activity or concurrent executions between previous and current activities) do rarely happen. But these are not explicitly described in Figure 1.

The development process consists of two successive phases, design phase and debug phase. One characteristic of the design phase is that the review activity is introduced after each design activity and coding activity. The design is divided into four stages: Concept, Function, Structure and Module. On the other hand, debug phase consists of the repetition of a pair of test and debug activities for four different objectives: Unit, Integration, Function and Verification.

3.2 Metrics for Software Development

In this paper, we adopt the following three kinds of metrics: SLC, FQ and TP [4][9] to analyze and evaluate the software development from the viewpoints of the quality of software products and the productivity of the team.

(1) Size of delivered code SLC

This metric counts the total lines of source codes including those reused, but excludes comments. Also, the lines of reused source code are calculated according to the degree of modification. Thus this metric is intended to evaluate the size of the final software products developed by the project.

We introduce the following six symbols:

SLC : size of delivered code

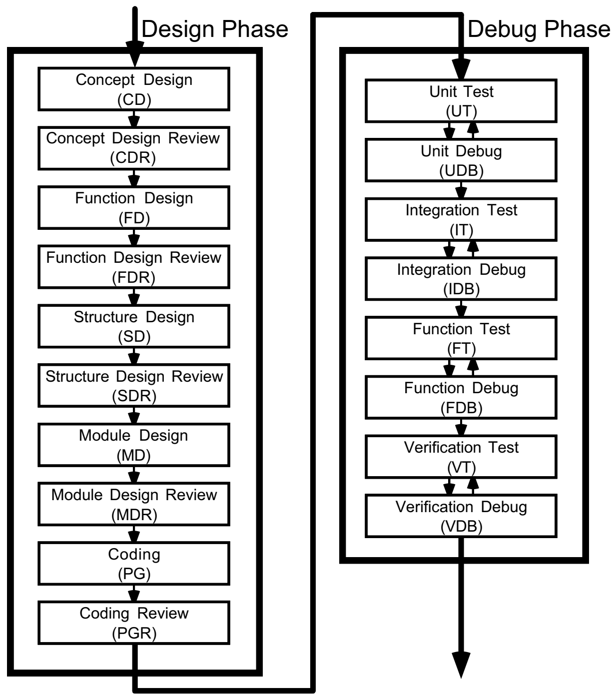

Figure 1: Process model

$newSLC$: size of code which was newly developed.

$slgSLC$: size of code which was modified slightly.

$extSLC$: size of code which was modified extremely.

The values of these symbols are measured by Kstep.

α, β : empirical constants.

Then the size of delivered code SLC is defined as follows:

$$SLC = newSLC + \alpha \times slgSLC + \beta \times extSLC$$

(2) Quality of delivered code FQ

This metric is defined by a normalized value of the number of faults detected during six months after the code delivery by the size of delivered code. Thus this metric is intended to evaluate the quality of the final software products developed by the project.

We introduce the following two symbols:

FQ : quality of the delivered code(measured by the number of faults/Kstep)

FD : the number of faults detected during six months after code delivery.

Then the quality of delivered code FQ is defined using FD and SLC as follows:

$$FQ = \frac{FD}{SLC}$$

(3) Productivity of the team TP

This metric is intended to evaluate the average productivity of all developers in the development team working for the project. Therefore, it is defined by the ratio of the size of delivered code on the total amount of efforts needed in the development.

We introduce the following two symbols:

TP : productivity of the team(measured by Kstep/person-month).

EFT : the total amount of efforts needed in the development (measured by person-month).

Then the productivity of the team TP is defined using EFT and SLC as follows:

$$TP = \frac{SLC}{EFT}$$

3.3 Metrics for Development Plan

(1) Cost estimation accuracy RE

Here, we use the person-month as the unit of cost rather than dollar or yen. This metric is intended to evaluate the appropriateness and feasibility of the development plan. Intuitively speaking, this metric is defined as the relative error of the cost estimated at the beginning of the project from the actual cost at the end of project.

Now, we introduce the following three symbols:

RE : cost estimation accuracy(measured by %).

$actCOST$: the actual cost(measured by person-month).

$estCOST$: the estimated cost(measured by person-month), which is determined in development plan.

Then the cost estimation accuracy RE is defined as follows:

$$RE = \frac{actCOST - estCOST}{estCOST} \times 100$$

(2) Faithfulness FI

This metric is intended to evaluate two kinds of faithfulness: (1) development plan was constructed faithfully to standards of good writing (specified by SEPG), (2) development itself was performed faithfully to the plan (constructed by development team). Therefore, the faithfulness FI is evaluated from the two viewpoints: faithfulness to standards of good writing $staFI$, and faithfulness to the plan itself $dynFI$.

(2.1) Evaluation of $staFI$

First, the faithfulness ($staFI$) of constructing the development plan is evaluated with respect to the following four components of the development plan:

(a) WBS (Work Breakdown Structure) [14]

(b) Organization charts of project

(c) PERT(Program Evaluation and Review Technique) charts[10]

(d) A list of software products to be developed

Now, we explain each component in more detail. Concerning WBS, SEPG makes an inquiry into the following points: (1) level of description : whether an activity in WBS is for 2 person-months, and (2) responsibility : whether a responsible person is described clearly for each activity. For the organization chart, SEPG confirms the correspondence between the organization in WBS and the content of organization chart. Concerning the PERT chart, SEPG investigates whether the restrictions (such as development period, effort and developers) are satisfied, and whether the critical path is described clearly. At the same time, SEPG must confirm the correspondence between WBS and PERT chart. Finally, SEPG confirms that all of the output product is specified for each activity.

(2.2) Evaluation of $dynFI$

Next, the faithfulness ($dynFI$) of fulfilling the plan is evaluated with respect to the following two viewpoints:

(e) Software review

(f) Progress management

By the previous empirical study, we have confirmed that the software review has high correlation with quality of software[13], and defined a metric, called the ratio of review effort(that is, the ratio of the efforts spent in review activity to the total efforts spent in development). SEPG uses the same metric to evaluate whether review activities are executed properly. Furthermore, SEPG confirms by interview with the developers whether progress management has been done.

Now, we introduce the following three symbols:

FI : the faithfulness of the development plan (measured by the grade point with $0 \leq FI \leq 100$).

$staFI$: the grade point evaluated by SEPG with respect to four components (a), (b), (c) and (d) in the plan (measured by the grade point with $0 \leq staFI \leq 50$).

$dynFI$: the grade point evaluated by SEPG with respect to (e) and (f) (measured by the grade point with $0 \leq dynFI \leq 50$).

Then, the faithfulness of the development plan is defined as follows:

$$FI = staFI + dynFI$$

4 DATA OF DEVELOPMENT PROJECTS

4.1 Property of Projects

As described in subsections 2.1 and 3.1, the target projects are the development of computer control systems with embedded software in Company A. The systems analyzed are classified into three categories: banking application, railroad application and business application.

Though we omit the details, such embedded software implements rather complex functions dealing with many sensors, actuators and control signals including various kinds of interrupts. Furthermore, since it is delivered in the form of LSI chips, modification of the faults after delivery is very expensive. Thus high reliability is especially required for the embedded software.

In Company A, the development of such software is executed concurrently with the design and development of system hardware. Hence it is necessary to manage the whole project. Generally, in such a project, the specification of the software product is strongly influenced by the restrictions of the hardware design.

However, in the case of Company A, modification of a specification will occur in some specific and limited areas of the product, such as the layout of a screen or the execution speed of the CPU. Fortunately, Company A can decide the interface to the hardware and can choose the operating system itself. As a result, the content of the requirement specification of the embedded software will be relatively stable and only changed in a very limited way. As we described in subsection 3.1, this is one of the main reasons why the standard waterfall model shown in Figure 1 is still effectively used in Company A.

The targeted thirty-one projects in this study are categorized into three groups:

(1) Six projects related to the banking system : ATM.

We represent and refer to the six projects by $PB_1, PB_2, \cdots, PB_6$.

(2) Twenty-four projects related to railroad system : automatic gate machine, ticket vending machine.

We represent and refer to the twenty-four projects by $PR_1, PR_2, \cdots, PR_{24}$.

(3) One project related to retail system : POS terminal.

We represent and refer to it by PS_1.

The cost (that is, development effort) of these thirty-one projects ranges from 18.8 person-months to 185 person-months. The average cost is 46.3 person-months.

4.2 Evaluation of Development Plan

We evaluate the metric "faithfulness" FI for the development plan. The evaluation was performed from the two district viewpoints: $staFI$ (construction phase) and $dynFI$ (execution phase). According to SEPG's judgement, five attributes are evaluated and then the grade points are summed up. As described in subsection 3.3, both of $staFI$ and $dynFI$ range from 0 to 50.

Although there are thirty-one projects available for analysis, only seventeen projects can be evaluated for their development plans. The main reason is that various kinds of defectiveness (such as missing data) occur on some development plans. Since a few years has already passed since these projects were completed, SEPG could not interview the actual developers of the project.

Table 1 shows the result of evaluation of development plans. In the evaluation, we use six banking projects $PB_1, \cdots, PB_6$, ten railroad projects $PR_1, \cdots, PR_{10}$ (out of twenty-four projects) and one retail project PS_1.

In Table 1, the grades A, B, C and D are introduced to clarify cause-effort relations between changes C_1 and C_2 (described in subsection 2.1). The following shows criteria for the grades:

$$\begin{array}{lcr} A & : & 75 \leq FI \leq 100 \\ B & : & 50 \leq FI < 75 \\ C & : & 25 \leq FI < 50 \\ D & : & 0 \leq FI < 25 \end{array}$$

From Table 1, we can see that there are two projects PB_2, PR_{10} with grade A, eight projects with grade B, seven projects with grade C. No project is with grade D.

4.3 Evaluation of Projects

Besides the development plan described in subsection 4.2, we have collected various data according to metrics defined in subsections 3.2 and 3.3. They include the size of delivered code SLC, the number of faults detected after delivery FD, the total effort needed in the development EFT, the estimated development cost $estCOST$ and the actual development cost $actCOST$. Based on these fundamental data, we calculate the value of key metrics, such as the quality of delivered code FQ, the productivity of the team TP and the cost estimation accuracy RE.

However, the values of these metrics are confidential(as already explained in Section 1), we cannot publish the values themselves in this paper. Table 2 shows only the skeleton of the data with respect to these metrics.

Table 1: Evaluation of the plan

Project Name	$staFI$	$dynFI$	FI	Grade
PB_1	12	35	47	C
PB_2	39	45	84	A
PB_3	18	40	58	B
PB_4	39	25	64	B
PB_5	19	35	54	B
PB_6	23	30	53	B
PR_1	26	30	56	B
PR_2	26	30	56	B
PR_3	26	30	56	B
PR_4	19	25	44	C
PR_5	32	25	57	B
PR_6	12	35	47	C
PR_7	30	20	50	C
PR_8	23	25	48	C
PR_9	12	25	37	C
PR_{10}	25	50	75	A
PS_1	12	35	47	C

5 ANALYSIS

In this section, we verify the assertions A_1, A_2 and A_3 described in subsection 2.2 using the the data in Tables 1 and 2 from the actual thirty-one projects. Before the verification, we classified the projects into three classes: C_0, C_+ and C_-, based on the value of RE. The criteria of classification is borrowed from internal criteria used in Company A. In Company A, ten percent is considered to be an important threshold for the evaluation of RE. That is, if $-10 < RE < +10$, then the project is successful with respect to cost estimation. We classify the projects as follows:

$$\begin{array}{lcc} C_0 & : & -10 < RE < 10 \\ C_+ & : & RE \geq 10 \\ C_- & : & RE \leq -10 \end{array}$$

Table 2: Skeleton of project evaluation

Project Name	*SLC* (Kstep)	*FD* (# of faults)	*EFT* (person-month)	*estCOST* (person-month)	*actCOST* (person-month)

RE (%)	*FQ* (# of faults/person-month)	*TP* (Kstep/person-month)

That is, the class C_0 includes the projects whose cost estimation accuracy ranges from -10% to $+10\%$, and the classes C_+ and C_- include the projects whose RE's are greater than $+10\%$ and less than -10%, respectively.

Table 3 shows the value of RE and the classification for each project. From Table 3, the class C_0 includes seventeen projects, the class C_- includes five projects, and the class C_+ includes nine projects, respectively.

5.1 Evaluation of Estimation Accuracy(A_1)

(1) Comparison between grade and class

First, we investigate the relationship between the the grades A, B, C and D by the faithfulness FI and the classes C_-, C_0 and C_+ by the cost estimation accuracy RE. Table 4 shows the resultant relationship only for the seventeen projects shown in Table 1. For example, the class C_0 includes two projects with grade A, four projects with grade B, and one project with grade C.

From Table 4, we see that all projects with grade A belong to the class C_0. This provides limited evidence that if the development plan was constructed faithfully to standards and the development was performed faithfully to the plan, then the cost estimation of the corresponding project is very accurate.

On the other hand, we can observe that the most of projects with grade C tend to belong to either the class C_- or the class C_+. This implies that if the development plan was not constructed faithfully to standards or the development was not performed faithfully to the plan, then the cost estimation is not accurate, neither.

(2) Further analysis of FI and RE

Next, we investigate to a greater extent the relationship between FI and RE. In this further analysis, we take projects from the same application, and consider ten projects $PR_1, PR_2, \cdots, PR_{10}$ (which appear both in Table 1 and Table 3).

Then, we calculate the correlation coefficient between the FI and RE for the selected ten projects $PR_1, PR_2, \cdots, PR_{10}$. The calculated values are shown as follows:

$$\textit{the correlation coefficient between } FI \textit{ and } RE = -0.47$$
$$\textit{the correlation coefficient between } staFI \textit{ and } RE = -0.60$$

The first result(-0.47) implies that there are some extent of correlation between the faithfulness of development plan FI and the cost estimation accuracy RE(though it is limited to the railroad related projects). Furthermore, the second result(-0.60) implies that there are relatively high correlation between the faithfulness to standards of good writing of the development plan $staFI$ and the cost estimation accuracy RE. As a result, we can conclude the assertion A_1 is proved affirmatively.

5.2 Quality of Delivered Code $FQ(A_2)$

Here, we analyze the assertion A_2 about the quality of delivered code using the test of statistical hypothesis. In the analysis we apply all data from the thirty-one projects shown in Table 3.

Now, for the test of statistical hypothesis, we define μ_0 to be the average of FQ's of all projects which belong to the class C_0. Similarly, we define μ_+ and μ_- to be the averages of FQ's of all projects in the classes C_+ and C_-, respectively.

(1) Classes C_0 and C_+

We define two hypotheses H_0 and H_1 for two classes C_0 and C_+. The level of significance α is chosen as 0.05.

$$\text{Null hypothesis} \quad H_0 : \mu_+ = \mu_0$$
$$\text{Alternative hypothesis} \quad H_1 : \mu_+ > \mu_0$$

The test statistic is calculated by

$$T(\bar{x}_0, \bar{x}_+, s_0^2, s_+^2) = \frac{\bar{x}_+ - \bar{x}_0}{\sqrt{\frac{s_+^2}{N_+} + \frac{s_0^2}{N_0}}} = 1.997$$

Table 3: Evaluation of cost estimation accuracy

Project Name	Estimation accuracy $RE(\%)$	Class
PB_1	-21	C_-
PB_2	6	C_0
PB_3	19	C_+
PB_4	-25	C_-
PB_5	3	C_0
PB_6	-17	C_-
PR_1	1	C_0
PR_2	-2	C_0
PR_3	-5	C_0
PR_4	-17	C_-
PR_5	14	C_+
PR_6	24	C_+
PR_7	1	C_0
PR_8	32	C_+
PR_9	21	C_+
PR_{10}	9	C_0
PR_{11}	26	C_+
PR_{12}	2	C_0
PR_{13}	-1	C_0
PR_{14}	0	C_0
PR_{15}	13	C_+
PR_{16}	9	C_0
PR_{17}	-29	C_-
PR_{18}	6	C_0
PR_{19}	6	C_0
PR_{20}	-1	C_0
PR_{21}	-3	C_0
PR_{22}	11	C_+
PR_{23}	-2	C_0
PR_{24}	0	C_0
PS_1	11	C_+

where $\bar{x}_0$, s_0 and N_0 are the average of FQ, the standard deviation of FQ and the number of sample projects in the class C_0, respectively. Similarly, $\bar{x}_+$, s_+ and N_+ are the average, the standard deviation and the number of sample projects in the class C_+, respectively. (As we have explained, the contract with Company A prohibits us from mentioning the values $\bar{x}_0$, $\bar{x}_+$, s_0 and s_+ calculated for the test.) Then the distribution of $T(\bar{x}_0, \bar{x}_+, s_0^2, s_+^2)$ is the t-distribution with the degree of freedom

$$m = \frac{(\frac{s_+^2}{N_+} + \frac{s_0^2}{N_0})^2}{\frac{s_+^4}{N_+^2(N_+-1)} + \frac{s_0^4}{N_0^2(N_0-1)}} = 14.6 < 15.$$

From the t-distribution with the level of significance $\alpha = 0.05$, the critical region becomes $R = t_{15}(0.05) = 1.753$. Then,

$$T(\bar{x}_0, \bar{x}_+, s_0^2, s_+^2) > R$$

Table 4: Relationship between FI and RE

		Classification by RE: C_-	C_0	C_+
Grade by FI	A	0	2	0
	B	2	4	2
	C	2	1	4
	D	0	0	0

holds. Thus, the null hypothesis H_0 is rejected.

This result implies that there is a significant difference on the quality of delivered code, FQ, between the projects in the class C_0(cost estimation accuracy ranges from -10% to $+10\%$) and the projects in the class C_+(cost estimation accuracy is greater than $+10\%$). Thus, for projects in the classes C_0 and C_+, the assertion A_2 is proved affirmatively.

(2) Classes C_0 and C_-

However, from the test of statistical hypothesis, it is shown that there is no significant difference on the quality of delivered code between the projects in the classes C_0 and C_-.

5.3 Productivity of Team $TP(A_3)$

Finally, we analyze the assertion A_3 about the productivity of the team using the test of statistical hypothesis. In this analysis we also apply all data from the thirty-one projects. For the test, we define μ_0 to be the average of TP's of all projects which belong to the class C_0. Similarly, we define μ_+ and μ_- to be the averages of TP's of all projects in the classes C_+ and C_-, respectively.

(1) Classes C_0 and C_+

We define two hypotheses H_0 and H_1 for two classes C_0 and C_+. The level of significance is chosen as 0.05.

Null hypothesis	$H_0 : \mu_+ = \mu_0$
Alternative hypothesis	$H_1 : \mu_+ < \mu_0$

The test statistic is calculated by

$$T(\bar{x}_0, \bar{x}_+, s_0^2, s_+^2) = \frac{\bar{x}_+ - \bar{x}_0}{\sqrt{\frac{s_+^2}{N_+} + \frac{s_0^2}{N_0}}} = 2.743$$

where $\bar{x}_0$, s_0 and N_0 are the average of TP, the standard deviation of TP and the number of sample projects in the class C_0, respectively. Similarly, $\bar{x}_+$, s_+ and N_+ are the average, the standard deviation and the number of sample projects in the class C_+, respectively. (In this case also, we cannot show the values $\bar{x}_0$, $\bar{x}_+$, s_0 and s_+ calculated in the proof by the same reason.) Then the distribution of $T(\bar{x}_0, \bar{x}_+, s_0^2, s_+^2)$ is the t-distribution with the degree of freedom $m = 14.6 < 15$. From the t-distribution with the level of significance $\alpha = 0.05$, the critical region becomes $R = t_{15}(0.05) = 1.753$. Then,

$$T(\bar{x}_0, \bar{x}_+, s_0^2, s_+^2) > R$$

holds. Thus, the null hypothesis H_0 is rejected.

This result implies that there is significant difference on the productivity of the team between the projects in the class C_0(cost estimation accuracy ranges from -10% to $+10\%$) and the projects in the class C_+(cost estimation accuracy is greater than $+10\%$). Thus, for projects in the classes C_0 and C_+, the assertion A_3 is proved affirmatively.

(2) Classes C_0 and C_-

However, from the test of statistical hypothesis, it is shown that there is no significant difference on the productivity of team between the projects in the classes C_0 and C_-.

6 CONCLUSION

In this paper, we have proved three interesting assertions A_1, A_2 and A_3 as the results of empirical research. Although the implications by these assertions themselves may not be new for academia people, they may become a driving force in the software developing company for promoting process improvement through (1) exhaustive collection of fundamental data, and (2) establishing some kinds of standards (mentioned in Section 1).

The main result of our empirical research are summarized as follows:

- If the development plan in a project is constructed carefully and faithfully to standards of good writing, then it is associated with more accurate cost estimates.
- If the cost estimates of a project are accurate, then the quality of the delivered code in the project is higher and the productivity of the team is also higher.

The future research work includes the following:

- Development of procedures (or algorithms) for evaluating the faithfulness $dynFI$ objectively.
- Detailed analysis about the projects in the classes C_+ and C_- to detect human factors.
- Investigation of method to feedback the analysis result to actual development.

ACKNOWLEDGEMENTS

Authors would like to thank Prof. Koji Torii of Nara Institute of Science and Technology and Dr. Shinji Kusumoto of Osaka University for their discussions and advises to our analysis in this paper.

REFERENCES

[1] Aoyama M. : "Agile software process model", Proc. of the 21th Annual International Computer Software and Applications Conference, pp.454–459 (1997).

[2] Basili V. R., Condon S. E., Emam K. E., Hendrick R. B. and Melo W. : "Characterizing and modeling the cost of rework in a library of reusable software components", Proc. of the 19th International Conference on Software Engineering(the 19th ICSE), pp.283–291 (1997).

[3] Cimitile A. and Visaggio G. : "A formalism for structured planning of a software project", International Journal of Software Engineering and Knowledge Engineering, Vol.4, No.2, pp.277–300 (1994).

[4] Fenton N. E. and Pfleeger S. L. : "Software Metrics : A Rigorous & Practical Approach", PWS Publishing (1997).

[5] Heemstra F. J. : "Software cost estimation", Information and Software Technology, Vol.34, No.10, pp.627–639 (1992).

[6] Humphrey W. S. : *Managing the Software Process*, Addison Wesley, Reading, MA (1989).

[7] Humphrey W. S., Snyder T. and Willis R. : "Software process improvement at Hughes Aircraft", IEEE Software, Vol.8, No.4, pp.11–23 (1991).

[8] Kusumoto S., Mizuno O., Hirayama Y., Kikuno T., Takagi Y. and Sakamoto K.:"A new software project simulator based on generalized stochastic petri-net", Proc. of the 19th International Conference on Software Engineering(the 19th ICSE), pp.293–302 (1997).

[9] Möller K. H. and Paulish D. J. : "Software Metrics : A Practitioner's Guide to Improved Product Development", IEEE Press (Chapman & Hall Computing) (1993).

[10] Poiaga L. : "Operations research in project management and cost engineering : An outlook for new operational developments", European J. Operational Research, Vol.41, No.1, pp.1–14 (1989).

[11] Podorozhny R. M. and Osterweil L. J. : "The criticality of modeling formalisms in software design method comparison", Proc. of the 19th International Conference on Software Engineering(the 19th ICSE), pp.303–313 (1997).

[12] Tanaka T., Sakamoto K., Kusumoto S. and Kikuno T. : "Improvement of software process by process visualization and benefit estimation", Proc. of the 17th International Conference on Software Engineering(the 17th ICSE), pp.123–132 (1995).

[13] Takagi Y., Tanaka T., Niihara N., Sakamoto K., Kusumoto S. and Kikuno T. : "Analysis of review's effectiveness based on software metrics", Proc. of the 6th International Symposium on Software Reliability Engineering, pp.34–39 (1995).

[14] Tausworthe R. C. : "The Work Breakdown Structure in software project management", Journal of Systems and Software, Vol.1, pp.181–186 (1980).

[15] Yourdon E. : *Death March : The Complete Software Developer's Guide to Surviving 'Mission Impossible' Projects*, Prentice Hall Computer Books (1997).