Statistical Analysis of Deviation of Actual Cost from Estimated Cost Using Actual Project Data

Osamu Mizuno, Tohru Kikuno

Graduate School of Engineering Science, Osaka University
1-3, Machikaneyama, Toyonaka, Osaka 560-8531, Japan

Katsumi Inagaki, Yasunari Takagi, Keishi Sakamoto

OMRON Corporation
Kusatsu, Shiga 525-0035, Japan

Abstract

This paper analyzes cause and effect of the deviation of actual cost (measured by person-month) from estimated cost for software development project. Although the obtained results themselves may not be new from the academic point of view, they may give a nice motivation for developers to join process improvement activities in the software company and thus become a driving force for promoting process improvement.

To be precise, we show that if the projects are performed faithfully under the well organized plan(that is, the plan is first constructed according to the standards of good writing and then projects are managed and controlled to meet the plan), then the deviation of actual cost from estimated one becomes small. Next we show statistically that the projects with small deviation of the cost estimate tend to achieve high quality of final product and high productivity of development team. In this analysis, actual project data on 37 projects at a certain company are extensively applied.

Keywords: software development project, project plan, deviation of cost estimate, quality and productivity, statistical analysis

1 Introduction

This paper describes empirical research on process improvement in a certain company, which we call Company **A** for convenience. In the Company **A**, the software engineering process group(SEPG) has been organized for seven years, and the SEPG tried to pervade the process improvement into their company. This study is a part of process improvement activities which the SEPG has done in 1998 for the developers in their company. In the software development project, at first, the size to be developed is estimated. Next a plan is constructed based on the estimate. Then the development starts according to the plan. If the project is performed exactly as the plan specifies, the project is regarded as successful project. However some projects inevitably result in so-called confused projects[8] or death march projects[17], in which actual cost exceeded the estimated cost by 50%. Therefore it is strongly desired by the SEPG to reduce the number of the confused projects.

In order to reduce the confused projects, many methods and guidelines have already been proposed. The famous methods such as COCOMO[3] and Function Point method[1] aimed to make the estimate accurate.Next to improve quality and productivity, the review activities are introduced to detect the defects in the early stage[12]. Then the references[8,13] insisted on the importance of constructing an appropriate plan and utilizing it during the development. But it is clear that a good method or guideline do not have any effect if they are utilized or applied inappropriately in the development field. In order to guide appropriate application, developers must be motivated to utilize them. Actually, Humphrey[8] said that motivated professionals or developers can strive for superior performance. Therefore, we should not enforce developers to apply a new method before they can understand its benefit and importance, and thus they are nicely motivated to do it.

In this paper, we take notice of the deviation of actual cost from the estimated cost and regard projects with large cost deviation as confused projects. We introduce a new metric DV which denotes the difference between actual cost and estimated cost. On the other hand, we (including the SEPG) guess that the construction of appropriate plan and its adherent execution is the key point to reduce confused projects. So in order to motivate developers to construct a good plan and to execute it, we show its benefit and importance. To sum up intuitively, we will show that "construction of appropriate plan and its adherent execution" is a useful approach to reduce the confused projects (with the large DV).

To be precise, we will show the following two propositions: (P_1) if the plan is constructed and executed adherently, the deviation of the cost estimate, DV, becomes small, and (P_2) if the deviation of the cost estimate, DV, is small, both the quality of the product and the productivity of the team are high. As mentioned before, many researchers have already pointed out that appropriate plan and its adherent execution are important for software development[8,13]. However, there are few studies which prove the effect of appropriate plan quantitatively by using actual development data. In this study we apply 37 project data obtained from actual development in Company **A** and show the correctness of propositions P_1 and P_2 by correlation analysis and test of statistical hypothesis.

As for the proposition P_1, it is very difficult to define what is a good plan and is much more difficult to construct a good plan. Thus we consider plans satisfying some standards(prepared for construction of the plan) as good plans. Hence we make a checklist for the development plan(The detail of the checklist is described in Section 3). Based on the checklist, we judge and evaluate the plan. For this purpose we define a new metric AD_{plan}, which indicates whether or not the plan is constructed adherently to the standard.

Next we should take notice of the execution of the plan. Ideally, developers perform the project exactly as the plan specifies. But actually various problems often disturb the development. For example, many defects are found in the test activity, and unexpected effort is needed to remove them. Thus we evaluate the execution of the plan from two points of view: (1) whether the project is managed using the plan, (2) whether the ratio of review effort to entire effort is large enough to avoid the confusion. For this purpose we define a new metric AD_{exec}, which indicates whether or not the development is performed adherently to the plan. Then we perform a correlation analysis between the evaluation of the plan $AD(= AD_{plan} + AD_{exec})$ and DV(the deviation of the cost estimate). The result of analysis shows that there are some extent of correlation between them.

As for the proposition P_2, any projects finished with lower actual cost than estimated are likely considered to be successful from economical point of view. However, from project manager's point of view, those projects never adhere to their development plans. In this line we evaluate the resultant effects of the deviation of the cost estimate. To be precise, we investigate the relationship of the deviation of the cost estimate on both the quality of the final product and the productivity of development team. In this analysis we classify the projects into two distinct classes using DV(the deviation of the cost estimate): C_C and C_S. C_S includes the projects with $DV <$ 10%, and C_C includes the projects with $DV \geq$ 10%. The test of statistical hypothesis confirmed that both the quality and the productivity of the projects in C_S are higher than those of in C_C(The level of significance is chosen as 0.05).

The rest of this paper is organized as follows: Section 2 describes the preliminaries of this study, target projects, process model and fundamental data. The metrics used in the analysis are defined in Section 3. The analysis for the deviation of the cost estimate is performed in Section 4. It is shown that the deviation of the cost estimate is small if the plan is constructed adherently(to the standard) and the project is performed or managed adherently(to the constructed plan). The analysis of effect of the deviation of the cost estimate on the quality and productivity is shown in Section 5. It is shown that in the project with small deviation of the cost estimate, the quality of the delivered code is high and the productivity of the development team is high. Finally, Section 6 summarizes this paper.

2 Preliminaries

2.1 Target projects

The projects targeted in this paper 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.

The 37 projects targeted in this study are categorized into three groups:

(1) 8 projects related to the banking system : ATM.
We represent and refer to them by $PB_1, PB_2, \cdots, PB_8$.
(2) 26 projects related to railroad system : Automatic gate machine, Ticket vending machine.
We represent and refer to them by $PR_1, PR_2, \cdots, PR_{26}$.
(3) 3 projects related to retail system : POS terminal.
We represent and refer to them by PS_1, PS_2 and PS_3.

The cost (that is, development effort) of these 37 projects ranges from 18.8 person-months to 185 person-months. The average cost is 46.3 person-months.

2.2 Process improvement

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

– Exhaustive collection of fundamental data[14,15].
– Establishing standards for activities.
  · Constructing the project plan.
  · Describing the development process[9,14].

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(but unfortunately is very difficult to estimate exactly[2,5]), 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.

Both the development managers and the software engineering process group(SEPG) are eager to know the causes of these improvements(especially, improvement on the quality and the productivity). This gives a strong motivation to the statistical analysis in this paper using the data from actual projects in Company **A**.

On the contrary, in Company **A** some projects also result in so-called confused project[8] or death march project[17]. Although these confused projects are exceptional and rare cases in Company **A**, the SEPG have to identify the causes of the confusion. This gives another motivation to the statistical analysis in this paper.

2.3 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 process model shown in Figure 1 is a standard waterfall model. As is described in subsection 2.1, 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 require-

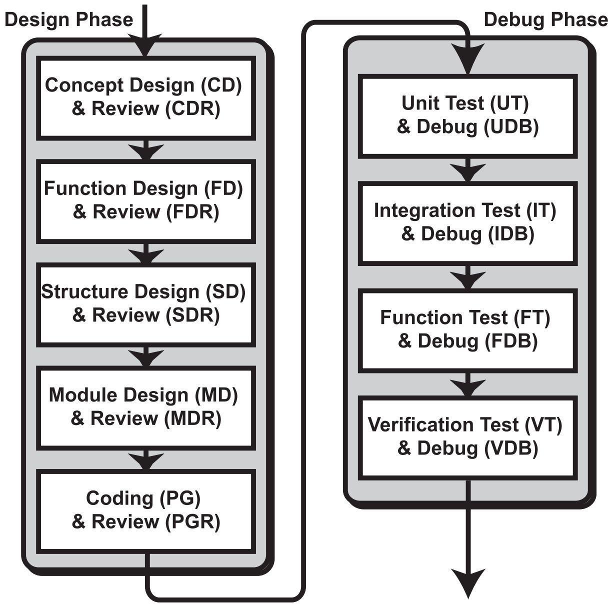

Fig. 1. Process model

ment 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.

2.4 Fundamental data

As a result of SEPG's activity, exhaustive but systematic collection of fundamental data from projects has been performed for several years. The collecting activity was executed according to software metrics recommendations[4,10,15], 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_3 and 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.

3 Definition of Metrics

In this Section, we introduce five kinds of metrics for the analyses to be described in Sections 4 and 5.

3.1 Metrics for analysis 1

(1) Deviation of the cost estimate DV

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

Now, we introduce the following three symbols:

DV : deviation of the cost estimate(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 deviation of the cost estimate DV is defined as follows:

$$DV = \frac{|actCOST - estCOST|}{estCOST} \times 100$$

(2) Adherence AD

This metric AD is intended to evaluate two kinds of adherence: (1) development plan was constructed adherently to standards of good writing (specified by the SEPG), (2) development itself was performed adherently to the plan constructed by development team. Therefore, the adherence AD is evaluated from the two viewpoints: adherence to standards of good writing, AD_{plan}, and development's adherence to the plan, AD_{exec}.

(A) Evaluation of AD_{plan}

First, the adherence of constructing the development plan (AD_{plan}) is evaluated with respect to the following four components of the development plan:

(a) WBS (Work Breakdown Structure)[16]
(b) Organization charts of project

(c) PERT (Program Evaluation and Review Technique) charts[11]

(d) A list of software products to be developed

Now, we explain each component in more detail. Concerning WBS, the 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, the SEPG confirms the correspondence between the organization in WBS and the content of organization chart. Concerning the PERT chart, the 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, the SEPG must confirm the correspondence between WBS and PERT chart. Finally, the SEPG confirms that all of the output product is specified for each activity. For each item described above, the grade points are given. Then the grade points are summed up as the metric AD_{plan}.

(B) Evaluation of AD_{exec}

Next, the development's adherence of executing the plan (AD_{exec}) 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[15], 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). The SEPG uses the same metric to evaluate whether review activities are executed properly. Furthermore, the SEPG confirms by interview with the developers whether progress management has been done. For each item described above, the grade points are given. Then the grade points are summed up as the metric AD_{exec}.

(C) Evaluation of AD

Now, we introduce the following three symbols:

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

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

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

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

$$AD = AD_{plan} + AD_{exec}$$

3.2 Metrics for analysis 2

As for the second analysis, we adopt the following three metrics: SLC, FQ and TP to analyze and evaluate the software development project from the viewpoints of the quality of software products and the productivity of the team.

(1) Size of delivered code SLC

This metric SLC 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

$newSLC$: 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 FQ 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 TP 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:

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

4 Analysis 1: Deviation of the Cost Estimate and Adherence

In order to prove the proposition P_1 in Section 1, we perform the first analysis. We analyze the deviation of the cost estimate(DV), and then we investigate the relationship between the adherence to the project plan(AD) and DV.

4.1 Assertion

Considering the metric which can be measured before a project begins, we have introduced the metric “adherence” in subsection 3.1. This metric is also considered as the quality of the development plan.

Here, we try to show the following assertion A_1 to investigate the relations between the development plan and the deviation of the cost estimate. The assertion A_1 is the same as the proposition P_1 in Section 1. For the analysis of A_1, we apply correlation analysis to actual data on 17 projects at Company **A**, which is the subset of the data shown in Table 1.

(A_1) In the project performed faithfully according to the project plan which is constructed adherently to standards, the deviation of the cost estimate is small.

The assertion A_1 implies that the adherence to standards in the construction of the project plan and the adherence to the project plan in the actual activities of the project are the main reasons for the small deviation of the cost estimate(that is, the difference between the actual cost and the estimated cost is small) of the project.

4.2 Classification based on deviation

We classify the projects into two classes: C_S and C_C, based on the value of DV. The criteria of classification is borrowed from internal criteria used by the SEPG in Company **A**. In Company **A**, ten percent is considered to be an important threshold and is actually applied for the evaluation of DV. At present the value of ten percent is determined empirically in Company **A** and is not yet validated theoretically [1] .

---

[1] The criteria of classification seems to be too restrictive compared with the one in death march project[17]. But in Company **A** for the projects which may result in confused

According to the criteria, the project, for which $0 \leq DV < 10$ holds, is determined to be successful with respect to the deviation of the cost estimate. On the other hand, the project, for which $DV \geq 10$ holds, is considered to be confused. Thus we classify the projects as follows:

$$C_S \;:\; 0 \leq DV < 10$$
$$C_C \;:\; DV \geq 10$$

That is, the class C_S includes the projects whose deviation of the cost estimate ranges from 0% to 10%, and the class C_C includes the projects whose deviation of the cost estimate are greater than 10%. These classes C_S and C_C can be considered as the class of successful projects and the class of confused projects, respectively, from the viewpoint of project cost.

Table 1 shows the value of DV and the classification for each project. From Table 1, the class C_S includes 19 projects and the class C_C includes 18 projects, respectively.

Table 1
Evaluation on the deviation of the cost estimate

Project Name	Estimation Accuracy		Project Name	Estimation Accuracy	
	DV(%)	Class		DV(%)	Class
PB_1	21.2	C_C	PR_{12}	1.7	C_S
PB_2	6.3	C_S	PR_{13}	0.8	C_S
PB_3	18.9	C_C	PR_{14}	0.2	C_S
PB_4	24.7	C_C	PR_{15}	13.2	C_C
PB_5	2.5	C_S	PR_{16}	8.6	C_S
PB_6	17.1	C_C	PR_{17}	29.1	C_C
PB_7	12.2	C_C	PR_{18}	6.4	C_S
PB_8	20.1	C_C	PR_{19}	6.2	C_S
PR_1	1.0	C_S	PR_{20}	0.8	C_S
PR_2	2.4	C_S	PR_{21}	3.2	C_S
PR_3	5.1	C_S	PR_{22}	10.7	C_C
PR_4	17.2	C_C	PR_{23}	2.3	C_S
PR_5	14.0	C_C	PR_{24}	0.0	C_S
PR_6	24.4	C_C	PR_{25}	0.0	C_S
PR_7	0.7	C_S	PR_{26}	8.2	C_S
PR_8	31.7	C_C	PS_1	11.3	C_C
PR_9	21.2	C_C	PS_2	24.5	C_C
PR_{10}	9.0	C_S	PS_3	23.9	C_C
PR_{11}	26.0	C_C			

4.3 Grading development plans

According to the assertion A_1, we evaluate the metric "adherence" AD for the development plan. The evaluation was performed from two distinct viewpoints: AD_{plan}

projects, reconstruction of the plan and introducing new resources are executed during the development. Thus the value of DV tends to be small even in the confused projects as shown in Table 1

Table 2
Evaluation of the plan

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

(construction phase) and AD_{exec} (execution phase). According to the SEPG's judgment, five attributes are evaluated and then the grade points are summed up. As described in subsection 3.1, both of AD_{plan} and AD_{exec} range from 0 to 50.

Although there are 37 projects available for analysis, only 17 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 after these projects were completed, the SEPG could not interview the actual developers of the project. In such projects, we cannot collect the data needed to evaluate the adherence AD.

Table 2 shows the result of evaluation of development plans. In the evaluation, we use 6 banking projects $PB_1, \cdots, PB_6$ (out of 8 projects), 10 railroad projects $PR_1, \cdots, PR_{10}$ (out of 26 projects) and one retail project PS_1 (out of 3 projects).

In Table 2, the grades $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ are introduced to clarify cause-effect relationship among the review process improvement. Intuitively speaking, the grades represent the degree of adherence at a glance, then they are easier to understand than the values of AD themselves. The following shows criteria for the grades:

$$\begin{aligned} \mathcal{A} &: \quad 75 \leq AD \leq 100 \\ \mathcal{B} &: \quad 50 \leq AD < 75 \\ \mathcal{C} &: \quad 25 \leq AD < 50 \\ \mathcal{D} &: \quad 0 \leq AD < 25 \end{aligned}$$

From Table 2, we can see that there are two projects PB_2 and PR_{10} with grade $\mathcal{A}$, eight projects with grade $\mathcal{B}$, and seven projects with grade $\mathcal{C}$. No project is with grade $\mathcal{D}$.

4.4 Grade vs. classification

First, we investigate the relationship between the grades $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ by AD and the classes C_S and C_C by DV. Table 3 shows the resultant relationship for the 17 projects shown in Table 2. For example, the class C_S includes 2 projects with grade $\mathcal{A}$, 4 projects with grade $\mathcal{B}$, and one project with grade $\mathcal{C}$.

From Table 3, we see that all projects with grade $\mathcal{A}$ belong to the class C_S. This provides limited evidence that if the development plan was constructed adherently to standards and the development was performed adherently to the plan, then the deviation of the cost estimate of the corresponding project is very small.

Table 3
Relationship between AD and DV

		Classification by DV	
		Cs	Cc
Grade by AD	A	2	0
	B	4	4
	C	1	6
	D	0	0

On the other hand, we can observe that most of projects with grade $\mathcal{C}$ tend to belong to the class C_C. This implies that if the development plan was not constructed adherently to standards or the development was not performed adherently to the plan, then the deviation of the cost estimate is relatively large.

4.5 Correlation between grade and deviation

Next, we investigate to a greater extent the relationship between AD and DV. In this further analysis, we take projects from the same application, and consider 10 projects $PR_1, PR_2, \cdots, PR_{10}$ (which appear both in Table 2 and Table 1).

Figure 2 shows the correlation between DV and AD for selected 10 projects. We calculate the correlation coefficient between the AD and DV:

$$\textit{the correlation coefficient between } AD \textit{ and } DV$$
$$= -0.47$$

The result(-0.47) implies that there are some extent of negative correlation between the adherence of development plan AD and the deviation of the cost estimate DV(though it is limited to the railroad related projects).

In order to find more strong result, we take AD_{plan} rather than AD and do the same analysis. The calculated value of correlation coefficient between the AD_{plan} and

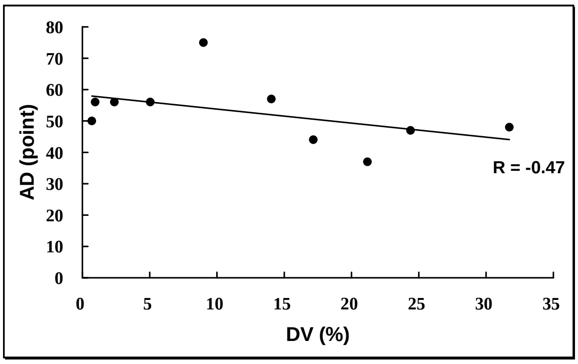

Fig. 2. Correlation between DV and AD.

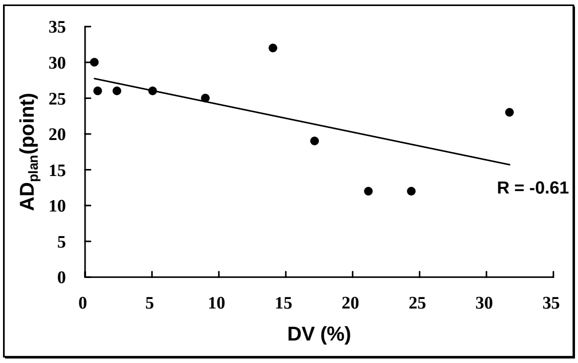

Fig. 3. Correlation between DV and AD_{plan}.

DV is shown as follows:

$$\textit{the correlation coefficient between } AD_{plan} \textit{ and } DV$$
$$= -0.61$$

The result(-0.61) implies that there is relatively high negative correlation between the adherence to standards of good writing of the development plan AD_{plan} and DV. As a result, we can conclude the assertion A_1 is proved affirmatively [2].

[2] In this analysis, we did not obtain strong relationship between DV and AD_{exec}. Therefore, AD_{exec} is one of the reasons which make the AD's correlation coefficient(=0.47) smaller than AD_{plan}'s(=0.61). However, this result does not imply that the project management is unnecessary. In Company **A**, since some management activities(that is, encouraging to perform the project according to the given plan) have already been done in all the projects, the differences among projects were not shown clearly by metric AD_{exec}.

5 Analysis 2: Effect of Deviation

We can see that the estimates of the project become accurate for these years in Company **A**. And we have also observed some improvements in both the quality and the productivity. In this Section, we clarify the relations between the deviation of the cost estimate and both the quality and the productivity.

5.1 Assertions

In order to analyze the relations between the deviation of the cost estimate and the quality, and the deviation of the cost estimate and the productivity, we introduce following assertions:

(A_2) In the project with the small deviation of the cost estimate, the quality of the delivered code is high.

The assertion A_2 implies that the project, for which the deviation of the cost estimate is small, delivers the final software product with high quality. As mentioned in subsection 2.4, 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 small deviation of the cost estimate, 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[17]).

5.2 Quality of delivered code

Here, we analyze the assertion A_2 which concerns the quality of delivered code using the test of statistical hypothesis. In the analysis we apply all the data from 37 projects shown in Table 1.

Table 4
The average values of FQ and TP

Class	Number of Projects	$FQ_{(normarized)}$	$TP_{(normarized)}$
C_C	19	1.00	1.00
C_S	18	3.23	0.56

The average of FQ's for each class is shown in Table 4. Since we cannot show the raw values by the contract with Company **A**, the values of both FQ and TP shown in Table 4 are normalized by the value of C_C.

Now, for the test of statistical hypothesis, we define μ_S^{FQ} to be the average of FQ's of all projects which belong to the class C_S. Similarly, we define μ_C^{FQ} to be the average of FQ's of all projects in the class C_C.

We define two hypotheses H_0 and H_1 for two classes C_S and C_C. The level of significance α is chosen as 0.05.

$$\text{Null hypothesis} \quad H_0 : \mu_C^{FQ} = \mu_S^{FQ}$$
$$\text{Alternative hypothesis} \quad H_1 : \mu_C^{FQ} > \mu_S^{FQ}$$

The null hypothesis H_0 is rejected by the t-test. Then the hypothesis H_1, that is $\mu_C^{FQ} > \mu_S^{FQ}$, holds statistically.

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

5.3 Productivity of team

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 37 projects. The mean value of TP's for each class is also shown in Table 4.

For the test, we define μ_S^{TP} to be the average of TP's of all projects which belong to the class C_S. Similarly, we define μ_C^{TP} to be the average of TP's of all projects in the class C_C.

We define two hypotheses H_0 and H_1 for two classes C_S and C_C. The level of significance is chosen as 0.05.

$$\text{Null hypothesis} \quad H_0 : \mu_C^{TP} = \mu_S^{TP}$$
$$\text{Alternative hypothesis} \quad H_1 : \mu_C^{TP} < \mu_S^{TP}$$

The null hypothesis H_0 is rejected by the t-test. This result implies that there is significant difference on the productivity of the team between the projects in the class C_S(the deviation of the cost estimate ranges from -10% to $+10\%$) and the projects in the class C_C(the deviation of the cost estimate is greater than $+10\%$ or less than -10%). Thus, for projects in the classes C_S and C_C, the assertion A_3 is proved affirmatively.

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 2).

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

– If a project is performed faithfully according to a good plan which is constructed carefully and adherently to the standards, then the deviation of the cost estimate(that is, the deviation of the actual cost from the estimated cost) of the project is small.
– If the deviation of the cost estimates of a project is small, then the quality of the delivered code in the project is higher and the productivity of the team is also higher.

The SEPG fed back these results to the developers in Company **A**.

The future research work includes the following:

– Development of procedures (or algorithms) for evaluating the adherence AD_{exec} objectively.
– Detailed analysis about the projects in the class C_C to detect human factors.
– Investigation of method to feedback the analysis result to actual development.

Acknowledgement

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

References

[1] A. J. Albrecht and J. E. Gaffney, Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation, *IEEE Trans. on Software Engineering* **9** (1983) 639–648.

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

[3] B. W. Boehm, *Software Engineering Economics* (Prentice-Hall, 1981).

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

[5] F. J. Heemstra, Software cost estimation, *Information and Software Technology* **34** (1992) 627–639.

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

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

[8] W. S. Humphrey, *A Discipline for Software Engineering* (Addison Wesley, Reading, MA, 1995).

[9] S. Kusumoto, O. Mizuno, Y. Hirayama, T. Kikuno, Y. Takagi and K. Sakamoto, A new software project simulator based on generalized stochastic petri-net, *Proc. of 19th Int'l Conference on Software Engineering* (1997) 293–302.

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

[11] L. Poiaga, Operations research in project management and cost engineering : An outlook for new operational developments, *European J. Operational Research* **41** (1989) 1–14.

[12] A. A. Porter, H. P. Siy, C. A. Toman and L. G. Votta, An experiment to assess the cost-benefits of code inspections in large scale software development, *IEEE Trans. on Software Engineering* **23** (1997) 329–346.

[13] I. Sommerville, *Software Engineering*, (Addison-Wesley, 1992).

[14] T. Tanaka, K. Sakamoto, S. Kusumoto and T. Kikuno, Improvement of software process by process visualization and benefit estimation, *Proc. of 17th Int'l Conference on Software Engineering* (1995) 123–132.

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

[16] R. C. Tausworthe, The Work Breakdown Structure in software project management, *Journal of Systems and Software* **1** (1980) 181–186.

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