Approach to the Technology that supports Hypothesis Testing for Software Cost Estimation

Fumiyoshi Sasagawa[1)], Toru Hanayama[2)], Akira Kawamura[1)], Terunobu Kume[1)], Takahiro Saito[1)], Jun Arima[1)] and Hiromi Kusumoto[2)]
1) FUJITSU LABORATORIES LTD. 2) FUJITSU LTD.
E-mail:f-sasagawa@jp.fujitsu.com

Abstract

The technology and the evaluation improvement technique for achieving the software cost estimation model that excluded the vagueness of project profiles as much as possible in cooperation with the PM Office were investigated. The model has a subdivided COCOMOII coefficient item, assumed binary (yes/no) input, and is expected to lead the development person-month effort for vague exclusion. Moreover, the risk level presumption technique to presume the evaluation of a COCOMOII coefficient item of seven values from a binary input automatically was developed for this achievement. In addition, an evaluation improvement technique consisting of the validity of the quality evaluation technology for the coefficient item by machine learning and the model, the accuracy assessments, and optimization approaches, etc. were developed. Based on these techniques, 63 project sample data received from a specialized estimation expert group or the PM Office, were analyzed, which lead to the extraction of the optimization model and candidates for improvement.

1. Introduction

Recent software development technology is sophisticated, specialized and complicated. Consequently, some projects have like amounts of development but, as follows the presence of a person with characteristic work experience, specification stability and difficulty of project development largely change project productivity. Prediction of profit and loss for the project is therefore difficult. Moreover, the way that relies on past experience and instinct doesn't lead to objectivity and transparent systematic technology advancements. The establishment of a software cost estimation technology that considers the project characteristic and the development characteristic is a pressing need.

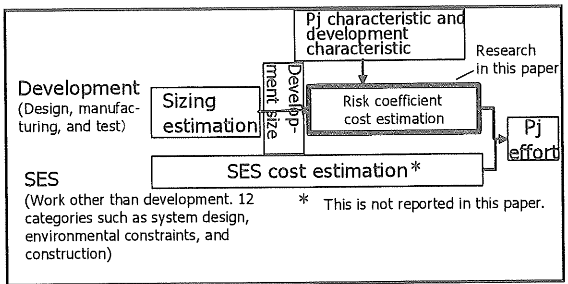


Figure 1. Dealings of several kinds of estimation

In this discussion, the cost estimation of software development is divided into,

a) Development (design, manufacturing and test) and b) SES (system design, environmental constraints and construction). In Development, the assessment technologies for "Sizing estimation" and "Risk coefficient cost estimation" are separately developed (see Figure 1). Our research relates to "Risk coefficient cost estimation."

The research is promoted in cooperation with the PM Office.

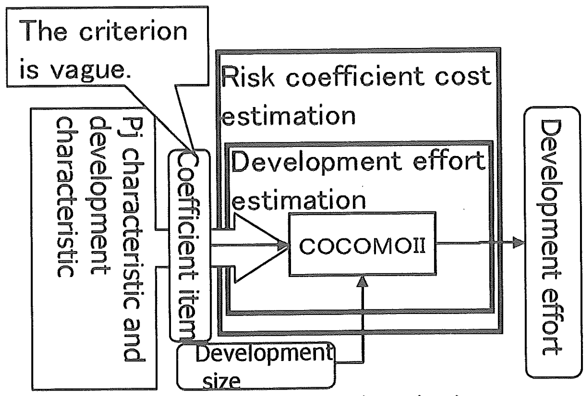


Figure 2. Conventional method

COCOMOII[1][2] is one of the most famous worldwide (see Figure 2) as a software cost estimation technology relating to risk coefficient cost estimation. COCOMOII is required for the model that outputs the development person-month effort and the development period to collect 13 input items of project data in each development environment, concerning the characteristics of the development size and project by seven-level assessment before actual use, and to adjust the parameters. However, because of the ambiguity of the criterion of seven levels (they were called coefficient items) and adverse effects of subjectivity, careful hearing of opinions by the experts is requires to collect high-quality project data.

Actually, it is a situation that is not able to correspond to actual use because of the cost of data gathering, and it is not being used now even though it is a famous cost estimation model.

2. An approach to the software cost estimation technology

To improve cost estimation technique, sophisticated cost estimation models and the PDCA cycle based on the feedback of evaluated on-site data are indispensable. However, conventional methods did not help homogenize the on-site data. Thus we fount it difficult to accumulate high-quality on-site data, and failed to turn the PDCA cycle (see ① of Figure 3) .

2.1. Establishment at PDCA cycle

The PM Office extracted the evaluation items that, we think, were related to the characteristic of the development software and the project to exclude vagueness as much as possible, and answered by binary (yes/no) as an objective event from failure project cases, etc. We developed the technology that automatically presumed an evaluation of coefficient items (seven levels of COCOMOII international standard software cost estimation technique, see ② of Figure 3). The binary evaluation item called "risk evaluation item" was developed instead of the coefficient item that was a COCOMOII input with ambiguity.

The PM Office asked approximately 100 project managers (PM) for the offer of projects, for the site data collection, for the construction of the model, (made in detail at the numeric level) and for the evaluation in the completion project (see ③ of Figure 3). This PM cost estimation data is summarized in Table 1. It is expected that PM can carry out the most competent judgment about the risk level of the development software product, the project characteristic and the coefficient items. Completion project became an object moreover because it was necessary to obtain the results value of the amount of development size and the development person-month effort.

All 63 sample projects have gathered data in the completion projects. The number of projects that were able to be used for all the development effort estimation models was 46 projects (see ④ of Figure 3) due to the loss of the development person-month

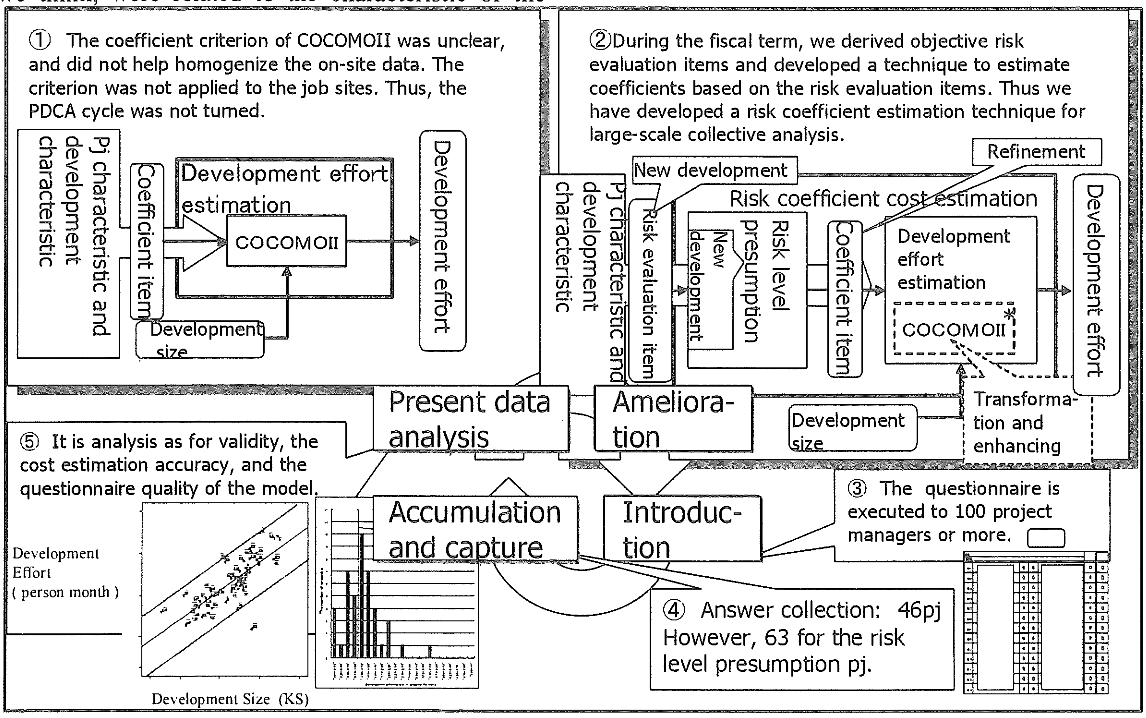


Figure 3. Main phases at PDCA cycle

effort, etc.

The evaluations of the structural validity of the model, the accuracy of the model, and the quality of each coefficient detail item were done based on the collected data (see ⑤ of Figure 3). As a result, we found that although the overall accuracy of this method was still inadequate at present , our approach was in the right direction, and that we have the possibility to establish the technology by turning the PDCA cycle. Furthermore, we specifically showed the improving guidelines of evaluation items to promote the efficiency of risk evaluation of users.

Thus, it is possible to do the analysis, and it is possible to make a tour of the PDCA cycle in the sample projects data.

2.2. Cost Estimation Model based on development risk

The estimation model developed in this paper leads to the development person-month effort from the risk evaluation item (all 175 items) and the development size (see Figure 4). This model divides into "Development effort estimation," where the development person-month effort is calculated from a coefficient detail item and development size, and "Risk level presumption," in which each seven-level value of "coefficient detail item" (25 items) is presumed from "Risk evaluation item".

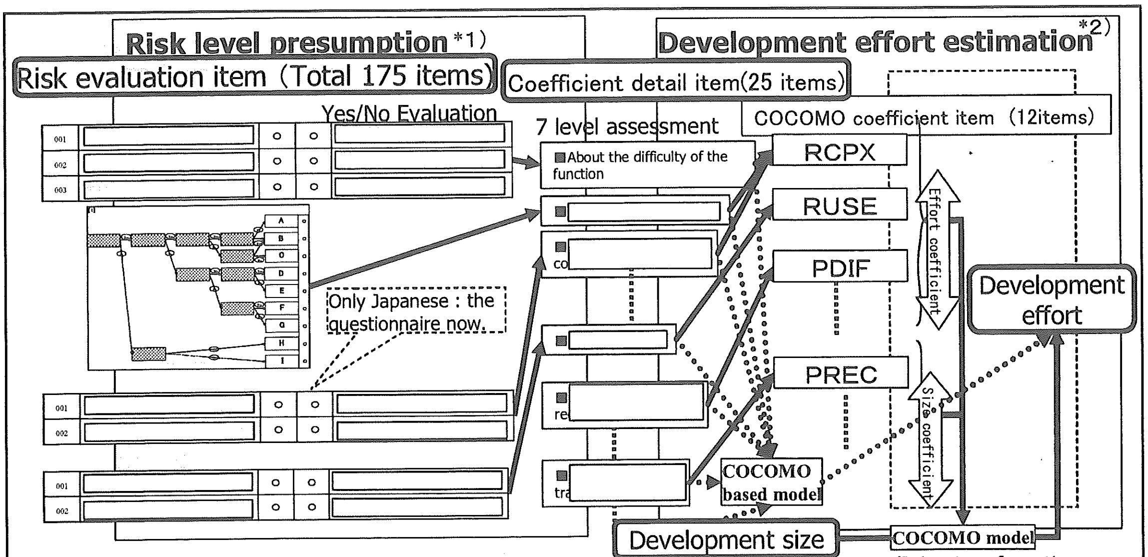
The sample data items collected from the project manager each become input to this "Risk level presumption" and "Development effort estimation" and output. Risk level presumption is a method to learn the relation between a binary vector of the risk evaluation item and the level of each coefficient detail item from sample data. It is possible to deal flexibly, without changing the algorithm, even when the risk evaluation item is changed. And we obtain the evaluation level at each region where the estimation is applied as a result that can be used only for the collection of data. The algorithm is named SVM (support vector machine), from which many of the most excellent results have been put out as a learning scheme in recent years.

The development effort estimation is transformed and enhanced based on COCOMOII. First of all, the COCOMO coefficient item (12 items) is expanded to the coefficient detail item (25 items). In development effort estimation, two main methods are tested. In one, the value of the COCOMO coefficient item is requested from the coefficient

Table 1. Summary of the PM cost estimation sample data

Number of obtaining Pj data	46 (63 for risk level presumption)		
Number of risk evaluation item (total 175 items)			
Coefficient detail item	Numnber of risk evaluation item	Coefficient detail item	Numnber of risk evaluation item
Functionality	13	Development experience	5
Document	*	Platform experience	3
External connectivity	9	Project management	12
Specialty	6	Hardware reliability	5
Security	9	Software reliability	5
Complexity	9	Development infrastructure	8
Allowance	6	End-user involvement	11
Reusability	16	Team building	10
Infrastructure requirement	5	Objective sharing	8
Diffusion of transaction	*	Worker composition	*
End-user Requirement	8	Quality control	6
Specification lock-in	8	Risk	8
Work experience	5		

*: Special form (decision tree form etc.)



*1) New development
(The machine learning technique is applied.)

*2) It is an improvement for COCOMOII. (It is a transformation extension in the coefficient detail item base.)
Development effort=A× (development size)^ (B+0.1Σ(Size coefficient)) ×Π(effort coefficient)

Figure 4. A coordinated model of the risk level presumption and the development effort estimation

detail item, the COCOMOII model is adjusted, and the mapping value of the COCOMO coefficient item is the simple average value of the coefficient detail item of the corresponding COCOMO coefficient items. The other one enhances the COCOMOII model, and leads directly to the development person-month effort from the coefficient detail items and the development size.

3. Software Cost Estimation model evaluation by PM cost estimation sample data

The quality evaluation of the newly developed cost estimation model accuracy assessment, which is based on the PM cost estimation sample data, is investigated.

3.1. Risk level presumption

For accurate software cost estimation, it is necessary to consider various risks which can influence the productivity of the project. In this paper, the answer to the question (risk evaluation item) is grouped into a classification of those risks as a coefficient detail item, and an influence at the risk level to evaluate each risk is obtained from the results of questionnaire answers.

However, the risk level given here is based on a person's subjective answers, and not on a unified viewpoint that considers all results of the questionnaire. On a unified viewpoint, the function to suggest the risk level, as calculated from the answer result of the risk evaluation item, is needed to solve this problem. Then, the experiment that presumed the 7-level assessment of coefficient detail items from the answer results of corresponding risk evaluation items was conducted.

This tool is a classification engine based on the machine learning technique that is called SVM (Support Vector Machine). There is a detailed explanation in the reference. This result of the questionnaire was considered to be a correct answer, though the correct answer was needed in the machine learning.

We examined a closed test in this paper because the PM estimation sample data was insufficient to do a cross-validation test. In the closed test, all the correct answer data is used as training data and the test data are estimated, but it is the same as training data. It is expected that the performance in the ideal situation, when sufficient correct answer data is collected, can be estimated in a closed test.

The first place correct answer rate of the estimated result of each coefficient detail item is brought together in Table 2. The first place correct answer rate is to evaluate whether a risk level judged to have

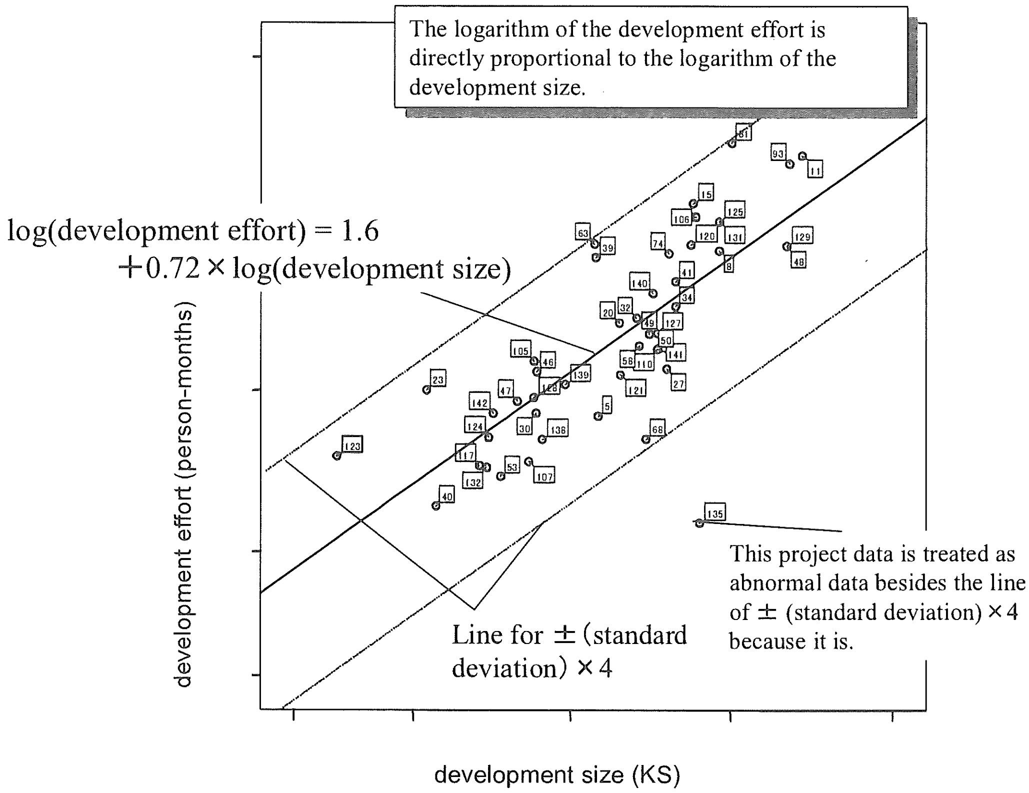


Figure 5. Log linearity of development person-months effort vs. development size

the highest possibilities by a classification engine is a correct answer.

Table 2. Risk level presumption result

Coefficient detail item	Maximum correct answer rate	Correct answer rate	Coefficient detail item	Maximum correct answer rate	Correct answer rate
1.Functionality	0.93	0.86	12.Development experience	0.81	0.74
2.External connectivity	0.89	0.70	13. Platform experience	0.60	0.60
3.Specialty	0.92	0.72	14.Project management	0.88	0.81
4.Security	0.86	0.76	15.Hardware reliability	0.73	0.73
5.Complexity	0.91	0.82	16.Software reliability	0.72	0.61
6.Allowance	0.73	0.58	17.Development infrastructure	0.88	0.76
7. Reusability	1.00	0.98	18. End-user involvement	0.95	0.90
8. Infrastructure requirement	0.80	0.68	19.Team building	0.82	0.73
9. End-user requirement	0.90	0.75	20. Objective sharing	0.87	0.80
10.Specification lock-in	0.78	0.65	21.Quality control	0.95	0.78
11.Work experience	0.66	0.57	22.Risk	0.89	0.69

[shaded]: Coefficient detail item for which the correct answer rate is more than 0.8.

Consequently, the correct answer rate accuracy becomes 73.7% on the average.

It is thought that the difference of this correct answer rate mainly depends on the number of risk evaluation items. The concept of the maximum correct answer rate is then introduced as a new criterion, and a unified evaluation is done on each item based on the viewpoint of how near the presumption result of the tool is to the maximum correct answer rate.

The maximum correct answer rate is the correct answer rate when the risk level with most numbers in one answer pattern is selected. The upper limit of the maximum correct answer rate is 1 when the answer pattern in the correct answer data has no overlap at all or when only one risk level is given in the same answer pattern even if there is overlap. As the amount of contradicted data in which two or more

Table 3. Lack risk evaluation items evaluated by entropy

Coefficient detail item	Entropy	Average	Coefficient detail item	Entropy	Average
1.Functionality	1.13944	0.087649	12.Development experience	1.68341	0.336681
2.External connectivity	1.23227	0.136919	13. Platform experience	1.34774	0.449246
3.Specialty	0.845916	0.140986	14.Project management	1.6667	0.138889
4.Security	1.26471	0.140523	15.Hardware reliability	1.43505	0.287009
5.Complexity	0.881613	0.097957	16.Software reliability	1.77619	0.355237
6.Allowance	1.87041	0.311735	17.Development infrastructure	1.3038	0.162975
7. Reusability	0	0	18. End-user involvement	0	0
8. Infrastructure requirement	1.56187	0.312374	19.Team building	0.77897	0.077897
9. End-user requirement	0.992216	0.124027	20. Objective sharing	0.992488	0.124061
10.Specification lock-in	2.43929	0.304911	21.Quality control	0.212412	0.035402
11.Work experience	2.06752	0.413505	22.Risk	0.499736	0.062467

Entropy: Number of minimum evaluation criteria that is necessary to presume complete

Average: Ambiguous in current state of about one evaluation criteria

[shaded]: Examination necessary group that ambiguous is high

risk levels are given to the same answer pattern increases, the maximum correct answer rate decreases. The maximum correct answer rate of each coefficient detail item is described in Table 2. The items in which the ratio (correct answer ratio) of the maximum correct answer rate and the correct answer rate of the classification engine tool is below 0.8 only number four. The average correct answer ratio is 0.878, and a high level of reliability is achieved.

By the entropy of coefficient detail items, we can consider how many risk evaluation items are needed to presume completely (see Table 3). Using these results, we can consider the quality of coefficient detail items' content. The entropy is defined by the equation $S=-(1/\log 2)\,\Sigma_i\,\Sigma_j P_{ij}\log P_{ij}$. Here, P_{ij} is the probability that risk level j is selected in answer pattern i and PlogP=0 at P=0 is assumed. The reason it is divided by log2 is because one answer to the risk evaluation item (question) is binary, of yes/no.

Entropy tends to rise when the number of questions is small. First of all, the value divided by the number of questions is calculated to find the coefficient detail item where entropy is large and the number of questions is small.

There is a large gap in the average entropy, 0.20-0.25. It is assumed that, for the questions of which the average entropy is exceeded, this gap should be examined, including the addition of questions.

3.2. Development effort estimation

The parameter of the COCOMOII model adopted as a base of the software cost estimation model is finally adjusted by two or more estimation specialists conferring after statistically processing the input data, and the risk level criterion is adjusted. The adjustment and accuracy assessment of the software cost estimation model parameters are done here by the statistical work that uses the PM cost estimation sample data result of the questionnaire.

The following three estimation models are evaluated based on data from which outliers have been removed and the evaluations are compared; **a)** "Linear model" which assumes that development size is only one factor of the simple model as an object of examining the validity of COCOMOII basic assumptions; **b)** "COCOMO model" which assumes COCOMO coefficient item to be factor as approximate COCOMOII model; and **c)** "COCOMO based model" which makes better use of only the structure of the COCOMOII expression and assumes the principal components of coefficient detail items to be factors. The result of each detail is reported as follows.

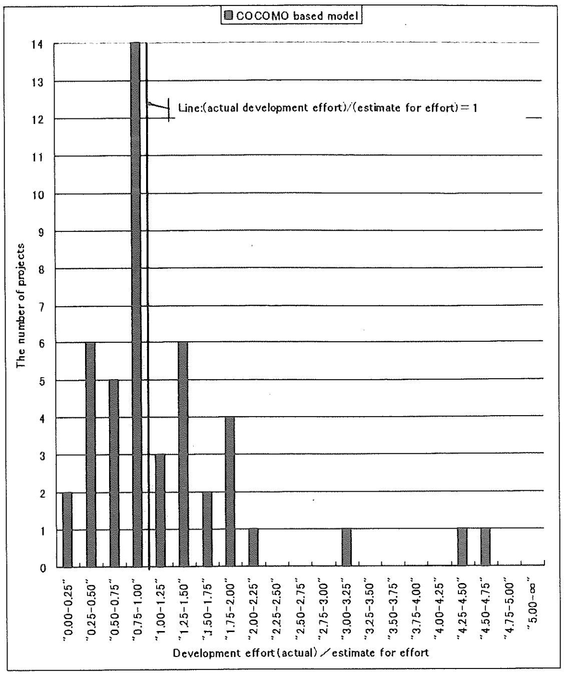


Figure 6. Histogram of (actual development effort) / (estimate for effort) for COCOMO based model

In "Linear model", project data with the development person-month effort of the remote value (results) by more than four times the standard deviation of estimation of the error is assumed to be an outlier from the value of the logarithm of the person-month effort obtained from the linear regression model.

Figure 5 shows a range that is four times the standard deviation of the model of a plot result of development size vs. development person-month effort of each project (results) and a linear regression result and the estimating errors. The estimation accuracy of each model after the parameter has been adjusted by using the entire data except outliers is summarized in Table 4. Here, the expression "PRED(.34) = 33%" means "Prediction accuracy: percentage of actual person-month efforts within 34% of the estimates". Structural validity of COCOMO II assumes that a base of both models is confirmed from the estimation accuracy of the COCOMO model, and that the COCOMO based model is better than the linear model. The COCOMO-based model's estimation equation is as follows.

Development effort = 5.61 × (Development Size) ^{E}
× (1st principal component of effort coefficient) $^{0.43}$
× (2nd principal component of e. c.) $^{-0.93}$
× (3rd principal component of e. c.) $^{0.0}$
× (4th principal component of e. c.) $^{0.56}$
× (5th principal component of e. c.) $^{0.65}$
× (6th principal component of e. c.) $^{-0.69}$

E=0.58+0.195 × (1st principal component of s. c.)
+ 0.104 × (2nd principal component of s. c.)
- 0.092 × (3rd principal component of s. c.)

Here, "e. c." is the abbreviation for "effort coefficient" and "s. c." is the abbreviation for "size coefficient".

Table 4. Prediction accuracy of each model (closed test)[1]

	PRED(.34)	PRED(.20)
Linear model	33%	22%
COCOMO model	42%	27%
COCOMO-based model	51%	36%

When we investigate how the development person-month effort (results) is distributed in the estimation of person-month effort to evaluate the accuracy of the model after adjustment, it became as shown in Figure 6.

4. Summary and discussions

The analysis evaluation of the validity of the questionnaire was done based on PM cost estimation sample data. In addition, the development effort estimation model in accordance with COCOMOII that calculated the development person-month effort was obtained, which is based on the PM cost estimation sample data. The validity of new technology to turn the PDCA cycle to efficiently refine the estimation technology was confirmed. It is necessary to develop the mechanism where the project manager on the site positively enters the information necessary for accumulation. By this mechanism, the PDCA cycle makes the estimation technology more useful.

5. References

[1] B.W. Boehm, C. Abts, A.W. Brown, S. Chulani, B.K. Clark, E. Horowitz, R. Madachy, D. Reifer and B. Steece, *Software Cost Estimation with COCOMO II*, Prentice HALL, New Jersey, 2000.

[2] B.W. Boehm, B. Clark, E. Horowitz, C. Westland, R. Madachy and R. Selby, "Cost models for future software life cycle processes: COCOMO 2.0", *Annals of Software Engineering 1*, Springer, Amsterdam, 1995, pp. 57-94.

[1] As the number of project data is increased, accuracy is expected to be high. We will report more accurate estimation technology in July, 2006.