Analyzing Risk Factors Affecting Project Cost Overrun

Masateru Tsunoda, Akito Monden,
Kenichi Matsumoto
Graduate School of Information Science
Nara Institute of Science and Technology
Nara, Japan
{masate-t, akito-m, matumoto}@is.naist.jp

Ryosuke Hatano, Toshihiko Nakano,
Yutaka Fukuchi
Hitachi, Ltd., Tokyo, Japan
{ryosuke.hatano.jn, toshihiko.nakano.yy,
yutaka.fukuchi.tt}@hitachi.com

***Abstract*—To prevent cost overrun of software projects, it is effective to predict the project which has high risk of cost overrun in the early phase of the project. In this paper, we clarify the risk factors which affect cost overrun. The risk factors are denoted by the questions such as "Are the customer's project goals clear?" The risk factors can be used as independent variables of the cost overrun prediction model. In the analysis, we used 290 projects' data collected in a software development company. The dataset was stratified by the project start time and the project size to eliminate their influence, and relationships between risk factors and cost overrun were analyzed with the correlation ratio. In addition, we focused risk factors which have strong and stable relationships to cost overrun, and analyzed them using the Sharpe ratio based index. As a result, we identified some risk factors which have relatively strong and stable relationships to cost overrun. After the analysis, we experimentally predicted cost overrun projects by collaborative filtering, using the risk factors as independent variables. The result suggested that cost overrun projects can be predicted by the risk factors.**

Keywords-correlation ratio; Sharpe ratio; stratification; risk management; collaborative filtering

I. INTRODUCTION

Recently, software is widely used as a part of infrastructure of the our daily life such as banking system and air traffic control system, while software size and cost (i.e. development effort) became extremely larger than ever. As a result, one single overrun project can cause serious damage to the profit of a software development company. Therefore, prevention of cost overrun became extremely important today.

One effective way to prevent cost overrun is to identify the project which has high risk of cost overrun (project failure) in the early phase of the project [4][10] so that countermeasures can be performed. To predict the project result (project failure), discriminant methods such as linear discriminant analysis or logistic regression has been used [4][10][13]. On a discriminant method, the project result is set as the dependent variable, and its value (i.e. cost overrun or not) is predicted from independent variables which are known at prediction point of time. Usually, project manager's answers for questionnaires related to risk factors (for example, the question is "Are the customer's project goals clear?" [6]) are used as independent variables for the project result prediction model [4][10][13]. The model is built from past projects' data, and current project's data is input as independent variables to predict the project result.

The objective of the paper is to clarify the risk factors which affect project cost overrun. They are used as independent variables of the cost overrun prediction model. We analyzed 290 projects' data collected in a software development company. In the dataset, each risk factor was evaluated by four-level Likert scale, and the degree of cost overrun was determined based on difference between estimated cost and actual cost. In the analysis, we examined relationships between each risk factor and cost overrun. When analyzing the dataset, we consider the change of characteristic of the dataset over time, because the analysis result shown in [2] suggests the characteristic varies over time.

After the analysis, we predicted cost overrun projects, using the factors which strongly relate cost overrun as independent variables. We applied collaborative filtering to predict cost overrun projects. It is originally used for the item (books or music) recommender system. Collaborative filtering is based on *k*-nearest neighbor algorithm, as the analogy based estimation method [9]. Roughly speaking, collaborative filtering finds projects similar to the target project, and makes prediction based on values of dependent variable of similar projects.

Our analysis results clarify which risk factors should be cared especially. Controlling the factors will suppress the probability of project cost overrun. Additionally, the results enable project managers to predict cost overrun projects. In what follows, Section II explains the dataset used in the analysis. Section III describes the analysis of relationships between risk factors and cost overrun. Section IV shows the results of cost overrun project prediction. Section V introduces related works. In the end, Section VI concludes the paper with a summary.

II. DATASET

We used risk evaluation data collected in a software development company in the 2000s. In the dataset, evaluations of risk factors and the degree of cost overrun is recorded for each software development project. The projects mainly developed enterprise application software. The

TABLE I. RISK FACTORS USED IN THE ANALYSIS

Identifier	Knowledge area	Description [6][7][8]
Upstream H10	Communication	Are there minutes of reviews with the customer, and have they been agreed upon (approved) by means such as approval signatures?
Upstream O	Cost	Is effort estimated by the quantitative estimation tool?
Downstream H24	Cost	How far can cost constraints be adjusted?
Upstream H46	Cost	Has the size of systemization been estimated? In doing so, has the basis for the estimate been recorded?
Upstream H1	Customer	Are requirements from customers of what they want to achieve clearly described in RFPs, etc? Also, have the project members understood them?
Upstream H72	Customer	If there is a need to assure current system functions, are current documents sufficiently maintained?
Upstream S3	Customer	Are the customer's project goals clear?
Upstream H28	Human Resources	Have key personnel with required business knowledge been acquired?
Upstream S15	Integration	Are the deliverables and products for each task clear?
Upstream H14	Organization	Does the project manager have experience appropriate for the scale and characteristics of the project? If the project manager's experience is insufficient, is there organizational support?
Upstream S7	Organization	In the project organization, are the responsibility assignment of stakeholders including customers clear, and are there any organizational deficiencies or concerns?
Downstream H54	Risks	What is the project size?
Upstream S14	Scope	Has the feasibility of the requirements defined in the specifications been verified?
Upstream H71	Scope	If the system connects to other systems developed by another company, is the responsibility assignment clear?
Upstream H42	Scope	Is there an agreement with the customer regarding the (contractual) handling of specification changes, and are measures such as separate payment being practiced?
Midstream H12	Technology	When using new programming language or technology, has the past record been confirmed on the application area? If there is no record, do you have the contingency plan for problems?
Upstream S11, S12	Human Resources, Time	Are there sufficient human resources with required skills? Has a plan for allocating personnel (in terms of quantity) been created?

dataset includes 290 projects and over 200 risk factors. The risk factors were evaluated by a project manager at a certain time such as the end of the design phase. We selected risk factors which were evaluated until acceptance of order, because we assume cost overrun prediction is performed at the time. Additionally, we only chose risk factors which are almost same as the factors which are already known to the public (defined in [6][7][8]). This is because risk factors are industrial secrets. As a result, we analyzed 17 risk factors described in Table I. In the table, each identifier corresponds to the identifier of the factors defined in [6][7][8]. The knowledge area denotes the area of PMBOK (Project Management Body of Knowledge) [5] to which each factor is classified (The classification is written in [6][7][8]). Note that only "Upstream O" is not defined in [6][7][8].

Each risk factor was evaluated by four-level Likert scale. The levels are "high risk," "middle risk," "low risk," and "unrelated". If status of a project corresponded with a description of a risk factor well, the risk factor was evaluated as "high risk." Similarly, if it did not correspond with the factor at all, the factor was evaluated as "low risk." When condition of a project was different from a risk factor, the factor was evaluated as "unrelated." For instance, when a system did not connect to other systems developed by another company, the risk factor "Upstream H71" was evaluated as "Unrelated." Some risk factors were evaluated by three-level Likert scale ("high risk," "low risk," or "unrelated").

Before analyzing, the evaluations of the risk factors ("high risk", "middle risk", "low risk", and "unrelated") were converted to numerical values (4, 3, 2, and 1). Some risk factors have missing values (i.e. a factor was not evaluated). The evaluations of the risk factors are originally used for project management (They are not used for cost overrun prediction).

Cost overrun is defined as the difference between estimated cost and actual cost. It is signified by six ranks (1 to 6), and small value means the difference was small (We do not know actual difference between estimated cost and actual cost, because they were not provided due to confidentiality). Note that cost overrun does not relate to profit well, because the profit is defined as the difference between price and cost, and it is different from each project.

In the analysis, projects whose cost overrun was greater than four were treated as cost overrun projects, and other projects were treated as non cost overrun projects. We assigned the value 1 to the cost overrun projects, and the value 0 to the non cost overrun projects. For discriminant methods such as linear discriminant analysis are used to predict cost overrun projects. Cost overrun projects are fairly fewer than non cost overrun projects (Although cost overrun projects defined in the paper are not failure project, we do

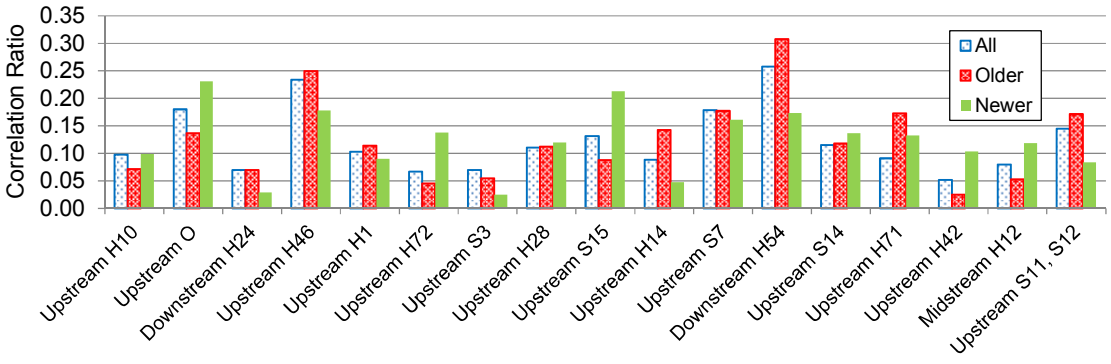

Figure 1. The correlation ratio of risk factors stratified by the project star time.

not show the actual rate of cost overrun projects because of confidential).

III. Relationships between Risk Factors and Cost Overrun

A. Analysis stratified by the project start time

We analyzed relationships between risk factors and cost overrun using the correlation ratio. As described in Section II, risk factors were treated as ordinal scale variable and cost overrun was treated as nominal scale. The correlation ratio is used to analyze the relationship between a nominal scale variable and an ordinal scale. The value range of the correlation ratio is [0, 1], and the large value indicates there is a strong relationship between the variables. The correlation ratio η^2 is calculated by:

$$\eta^2 = \frac{SST}{SSB} \quad (1)$$

In the equation, *SSB* is the sum of squares between groups (signified by the nominal scale variable), and *SST* is the sum of squares total.

We divided the dataset into two subsets in chronological order (based on the project start time), and calculated the correlation ratio on whole dataset and the two subsets. The analysis result shown in [2] suggests the characteristic varies over time. Similarly, we assumed the characteristic of the dataset was changed over time, because process improvement was performed during collecting the dataset, and it might affect the relationships between risk factors and cost overrun.

Figure 1 shows the correlation ratio of each risk factor stratified by the project start time. In the figure, "All," "Older," and "Newer" signify the correlation ratio on the whole dataset, the older subset, and the newer subset. On average, the strength of the relationships between the risk factors and cost overrun is not different between the older subset and the newer subset (Both averages were 0.12). However, some risk factors have large difference. So, the relationships between the risk factors and cost overrun were greatly changed over time. For example, the correlation ratio of Upstream S15 is small on the older subset, but it is large on the newer subset. On the contrary, the relationship between Downstream H54 and cost overrun weakened on the newer dataset. The result suggests the (learning) dataset should be divided when cost overrun projects are predicted.

B. Analysis stratified by the project start time and the project size

In Figure 1, Downstream H54 has the highest relationship to cost overrun on whole dataset. This means when the project size is large, the probability of cost overrun becomes high. There is a probability that only the project size (Downstream H54) affects cost overrun, and other risk factors have spurious relationships to cost overrun. So we analyzed the relationships between risk factors and cost overrun when the influence of the project size was eliminated. We stratified the dataset by the evaluation of Downstream H54 (project size) and analyzed them.

Figure 2 shows the correlation ratio of each risk factor when projects are stratified by the project start time and the project size. In the figure, "Older-low" signifies the correlation ratio on the subset where projects were older and their project sizes were small (The evaluations of Downstream H54 were "low risk"). Others such as "Newer-High" are the same meaning. We omitted projects whose evaluations of Downstream H54 were "unrelated," because most of their risk factors do not have relationships to cost overrun. To make the figure more visible, we cut the bar of Upstream H72 on newer-high subset (The actual value is 0.71).

On some risk factors such as Upstream S3, the differences of the correlation ratio among subsets are relatively small. That is, they have steady relationships to cost overrun on any subsets. But other factors such as Upstream H28 have unstable relationships. The former is considered to be common relationships, but the letter is not.

C. Analysis based on the Sharpe ratio based index

Figure 2 signifies there are risk factors which have steady relationships to cost overrun and factors which have unstable relationships. We assumed that on the former factors, the average of the correlation ratio is large and the variance is small among the subsets.

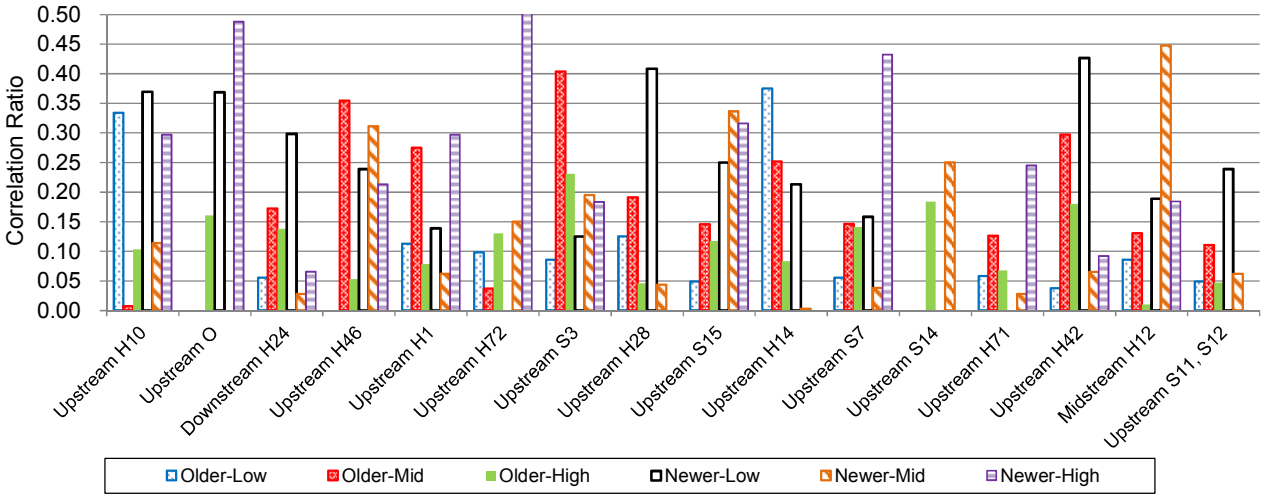

Figure 2. The correlation ratio of risk factors stratified by the project start time and the project size.

To identify the factors, we used the Sharpe ratio based index (SRBI) [11]. The Sharpe ratio is originally used to evaluate performance of a portfolio (combined financial products). It takes into account not only profit but also risk. The value is low when profit is high but risk (standard deviation) is also high. The Sharpe ratio s is calculated by:

$$s = (p - b) / r \quad (2)$$

where p is profit rate of a portfolio, r is standard deviation of the profit rate, and b is profit rate of risk-free asset. The SRBI c is calculated by:

$$c = (a - m) / d \quad (3)$$

where a is the average of the index, d is the standard deviation of it, and m is a baseline value whose function is same as b in equation (2). Originally, in equation (2), when p is smaller than b, the portfolio is regarded as useless. Similarly, in equation (3), when the target index was correlation coefficient and its value was 0.1, we considered it as meaningless and set m to 0.1.

Figure 3 illustrates the average, the standard deviation, and the SRBI of each risk factor. Top five risk factors which have large SRBI are Upstream H10, Upstream O, Upstream H46, Upstream S3, and Upstream S15. From the result, we concluded that they have relatively stable relationships to cost overrun. It is preferable to fulfill the following conditions to avoid cost overrun.

- There are minutes of reviews with the customer, and they have been agreed upon (approved) by means such as approval signatures. (Upstream H10)
- Effort is estimated by the quantitative estimation tool. (Upstream O)
- The size of systemization has been estimated. In doing so, the basis for the estimate has been recorded. (Upstream H46)
- The customer's project goals are clear. (Upstream S3)
- The deliverables and products for each task are clear. (Upstream S15)

D. *Analysis based on Precision*

We confirm the presence of the risk factor when its

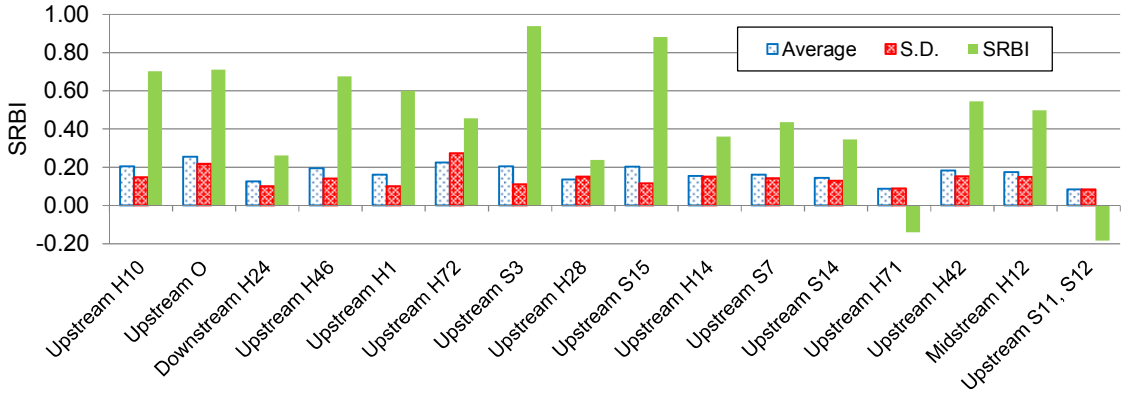

Figure 3. The SRBI of risk factors.

TABLE II. DEFINITIONS OF *TP*, *FN*, *FP*, AND *TN*

		Actual value	
		True	False
Predicted value	True	*TP*	*FP*
	False	*FN*	*TN*

evaluation is "high risk," the project will be cost overrun certainly. (Note that this does not mean when its evaluation is not "high risk," the project will be non cost overrun certainly). Also, we confirm the presence of the risk factor when its evaluation is "unrelated" or "low risk," the project will not be cost overrun certainly. The risk factors are useful to predict project result (cost overrun or not) manually.

To identify the factors, we used the precision. Originally, the precision is used to evaluate the accuracy of discriminant methods. The precision is calculated by:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

Definitions of *TP* (true positive), *FN* (false negative), *FP* (false positive), and *TN* (true negative) are denoted in Table II. When calculating the precision of a risk factor for cost overrun, "cost overrun" was treated as "The actual value is true," and "high risk" was treated as "The predicted value is true." Similarly, when calculating the precision of a risk factor for non cost overrun, "non cost overrun" was treated as "The actual value is true," and "unrelated" and "low risk" were treated as "The predicted value is true."

Additionally, for each risk factor, we confirm the rate that when a project was cost overrun, the factor of the evaluation was "high risk." We used the recall to calculate the rate. The recall is also used to evaluate the accuracy of discriminant methods. The recall is calculated by:

$$\text{Recall} = \frac{TP}{FP + FN} \tag{5}$$

Same as the precision, when calculating the recall of a risk factor toward cost overrun, "cost overrun" was treated as "The actual value is true," and "high risk" was treated as "The predicted value is true." Also, the calculation of recall of a factor for non cost overrun is same as the precision.

Based on the analysis result shown in Figure 1, we stratified the dataset by the project start time, and calculated the precision and the recall (We did not stratified the dataset by the project size because that makes the subset too small for calculating the precision and the recall).

Figure 4 shows the precision and the recall of each risk factor for cost overrun. As stated in Section II, cost overrun projects were fairly fewer than non cost overrun projects, and therefore the precision and the recall are not high. Although the precision of Upstream H46 is 100% on the older subset, the number of the project is only one. So, the result is not reliable. The precision of Upstream H1 and Upstream H71 is relatively high on the older subset. Four projects out of five were cost overrun when the evaluation of Upstream H1 was "risk high," and three projects out of four were cost overrun when the evaluation of Upstream H71 was "risk high." However, the precision is not high on the newer subset. So, we did not conclude they are notable risk factors.

Figure 5 shows the precision and the recall of each risk factor for non cost overrun. The precision and the recall are high on average, because cost overrun projects were fairly fewer than non cost overrun projects. There is no risk factor whose precision is greater than 90%.

In the analysis, we did not find the risk factor when its evaluation is "high risk," the project will be cost overrun certainly, and the factor when its evaluation is "unrelated" or "low risk," the project will not be cost overrun certainly.

IV. COST OVERRUN PREDICTION

A. Overview

We clarify prediction accuracy of cost overrun projects, using the risk factors which have relatively strong relationships to cost overrun. We used the risk factors whose correlation ratio was equal to or greater than 0.1. We predicted cost overrun projects using whole dataset and two

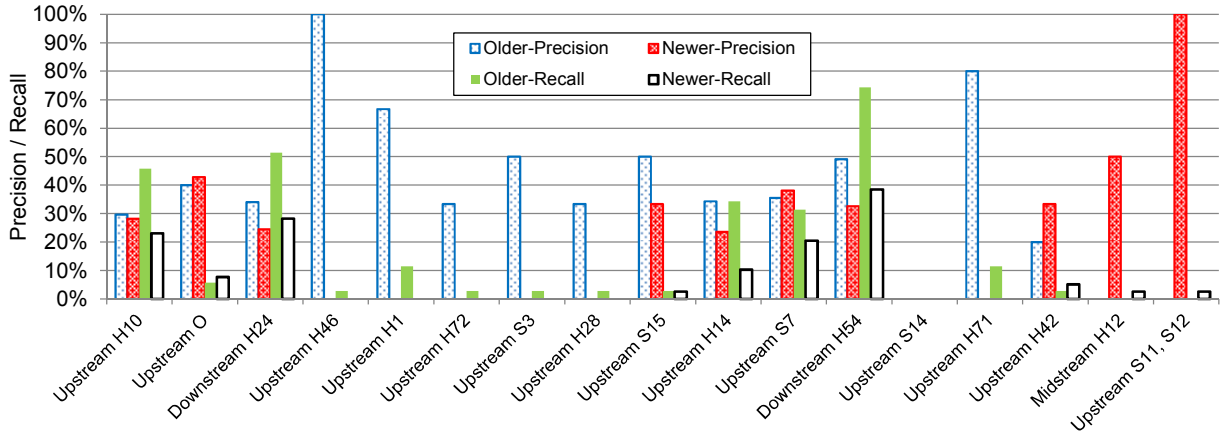

Figure 4. The precision and the recall of each risk factor for cost overrun

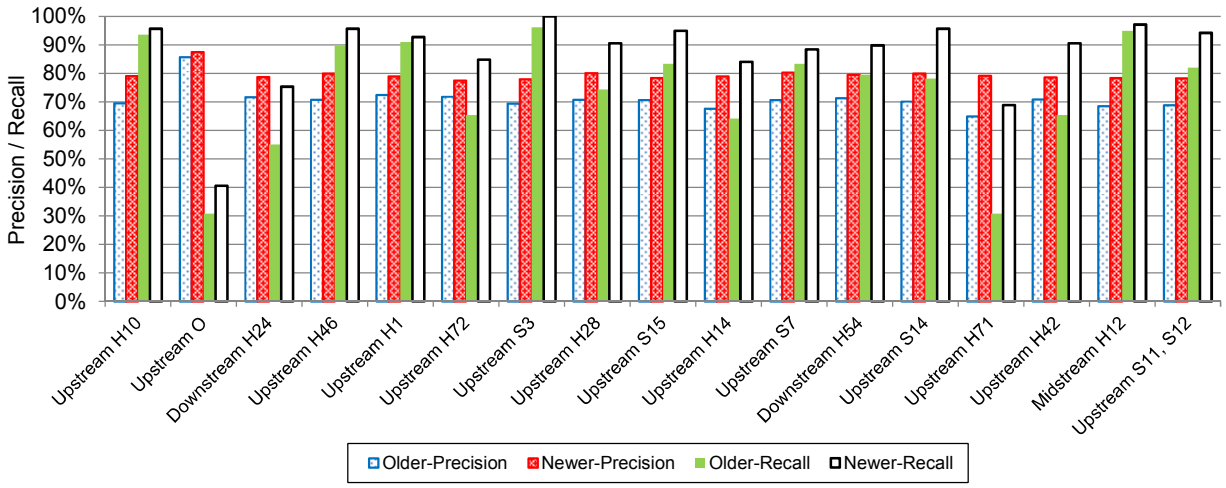

Figure 5. The precision and the recall of each risk factor for non cost overrun.

subsets of the dataset divided by project start time, and compared prediction accuracy of them. This is because the analysis result in Figure 1 suggests the dataset should be divided. We did not divide the dataset by the project size because it has strong relationship to cost overrun as shown in Figure 1. It means the risk factor of project size is effective to predict cost overrun. The other reason is that small subset depresses prediction accuracy.

We applied collaborative filtering for prediction, because when the percentage of cost overrun and non cost overrun projects are imbalanced, collaborative filtering is the most appropriate prediction method [12].

B. Collaborative filtering

Originally, collaborative filtering is used for the recommender system which estimates users' preferences to recommend items such as books or music. Collaborative filtering presumes "Users who have similar preferences like similar items."

Collaborative filtering uses $m \times n$ matrix shown in Table III. In the matrix, $Proj_i$ is *i-th* project, Q_j is *j-th* independent variable, v_{ij} is a value of Q_j of $Proj_i$, and y_i is the value of the dependent variable. We presume $Proj_a$ is predicted project, and $\hat{y}_a$ is the predicted value of y_a. Procedures of collaborative filtering consist of the three steps described below.

Step 1 (normalization): Since a dependent variable and independent variables have different ranges of value, this step makes the ranges [0, 1]. The value v'_{ij}, normalized the value of v_{ij} is calculated by:

$$v'_{ij} = \frac{v_{ij} - \min(Q_j)}{\max(Q_j) - \min(Q_j)} \tag{6}$$

In the equation, $\max(Q_j)$ and $\min(Q_j)$ denote the maximum and minimum value of Q_j respectively.

Step 2 (similarity computation): This step computes similarity $\mathrm{Sim}(Proj_a, Proj_i)$ between the predicted project p_a and other projects p_i by:

$$\mathrm{Sim}(Proj_a, Proj_i) = \frac{\sum_{h=1}^{m} v'_{ah} v'_{ih}}{\sqrt{\sum_{h=1}^{m} v'^{2}_{ah}} \sqrt{\sum_{h=1}^{m} v'^{2}_{ih}}} \tag{7}$$

The range of the value of $\mathrm{Sim}(Proj_a, Proj_i)$ is [0, 1].

Step 3 (computation of predicted value): The predicted value is computed by weighted average of the independent variable of similar projects. Formally, the predicted value is computed by:

$$\hat{y}'_a = \frac{\bar{v}'_a + \sum_{h \in Simprojects} \mathrm{Sim}(Proj_a, Proj_h)(y'_h - \bar{v}'_h)}{\sum_{h \in Simprojects} \mathrm{Sim}(Proj_a, Proj_h)} \tag{8}$$

In the equation, *Simprojects* denotes the set of k projects (*neighborhoods*) which have top similarity with $Proj_a$. The neighborhood size k affects prediction accuracy. The value $\hat{y}'_a$ is the normalized value of $\hat{y}_a$. The value $\bar{v}'_h$ is the average of v'_{ih} included in $Proj_h$. On the recommender system, collaborative filtering uses users' ratings for items. However, some people tend to rate every item as high, and

TABLE III. MATRIX USED BY COLLABORATIVE FILTERING

	Result	Q_1	Q_2	…	Q_j	…	Q_n
$Proj_1$	y_1	v_{11}	v_{12}	…	v_{1j}	…	v_{1n}
$Proj_2$	y_2	v_{21}	v_{22}	…	v_{2j}	…	v_{2n}
…	…	…	…		…		…
$Proj_i$	y_i	v_{i1}	v_{i2}	…	v_{ij}	…	v_{in}
…	…	…	…		…		…
$Proj_m$	y_m	v_{m1}	v_{m2}	…	v_{mj}	…	v_{mn}

on the other hand, some do as low. Hence, this equation uses the difference from average of each people's rating. We applied this algorithm to predict the project result, because our dataset seems to have similar characteristic.

In the experiment, we skipped Step 1 because the range of the value of each risk factor is the same as each other. The number of neighborhoods k was decided based on a preliminary analysis.

C. Evaluation criterion

We used area under the curve (AUC) [1] as the evaluation criterion of cost overrun prediction. AUC is recently used to evaluate discriminant methods in software engineering researches, for it is more appropriate criterion for discriminant methods than other criteria like F1 score [3]. The value range of AUC is [0, 1], and higher AUC means that prediction accuracy of the method is high. When AUC is smaller than 0.5, the prediction result is same as random prediction. AUC is defined as the area under the receiver operating characteristic (ROC) curve. ROC curve is drawn by changing threshold and calculating true positive rate and false positive rate. These rates are calculated by:

$$\text{True Positeve Rate} = \frac{TP}{TP + FN} \tag{9}$$

$$\text{False Positive Rate} = \frac{FP}{FP + TN} \tag{10}$$

High true positive rate and false positive rate means high accuracy. But there is tradeoff between them, and they depend on a threshold. For example, if prediction is done by logistic regression and the threshold is set as 0, true positive rate is very high but false positive rate is very low. AUC can evaluate performance of discriminant methods independently from the threshold.

D. Experimental Procedure

We predicted the project result (Cost will be overrun or not) according to the following procedure. The procedure was applied to whole dataset, older subset, and newer subset.

1. The risk factors whose correlation ratio is smaller than 0.1 are removed from the dataset.
2. In the dataset, a project is regarded as a test data (ongoing project and the project result is unknown), and other projects are regarded as the learning dataset (finished project and the project result is known).
3. The project result of the test data is predicted by collaborative filtering based on the learning dataset.
4. For each project in the dataset, Step 2 to 3 are repeated (leave-one-out cross-validation).
5. The evaluation criterion (AUC) is computed.

E. Prediction results

Table IV shows AUC on each dataset. To make comparison easy, predicting results based on whole dataset were divided by the project start time after the prediction. Prediction results based on the older subset and the newer subset show higher accuracy than whole dataset. Although the difference is not large, the result suggests that data stratification by the project start time is effective to enhance cost overrun prediction accuracy.

TABLE IV. AUC ON EACH DATASET

Learning data	AUC
Older subset	0.68
Newer subset	0.56
Whole dataset (Older; divided after prediction)	0.65
Whole dataset (Newer; divided after prediction)	0.54

AUC on each dataset is greater than 0.5, and therefore cost overrun projects can be predicted by the risk factors. However, AUC is not high, and other risk factors are needed to improve the prediction accuracy.

V. RELATED WORK

There are some researches which analyzed relationships between risk factors and project results, and predicted project results. Takagi et al. [10] defined confused projects based on the ratio of the actual resultant cost and the planned cost, and analyzed 32 projects collected in a software development company in 1990s. They pointed out risk factors about estimations and planning are important. Our analysis result and their result are similar in regard to the influence of Upstream S15. However, they did not analyze other risk factors which showed stable relationships in Figure 3 (Upstream H10, Upstream O, Upstream H46, and Upstream S3).

Procaccino et al. [4] defined success project based on asking of developers, and analyzed 42 projects collected from 21 developers in 1999. They identified some risk factors related to success of the project. The major difference between our research and their research is they did not analyze risk factors which showed stable relationships in Figure 3.

Also, Verner et al. [13] (co-author of [4]) analyzed relationships between success of the project (The definition is same as [4]) and risk factor. They collected datasets from the United States and Australia, and analyzed them. They pointed out the initial effort estimation is important for success of the project. However, they did not analyze risk factors which affect estimation accuracy. We analyzed the risk factors (Upstream O and Upstream H46), and that is the major difference between our research and their research.

VI. CONCLUSIONS

In this paper, we clarified risk factors which have relatively strong relationships to cost overrun of the software development project. We analyzed software development project dataset, removing effects of the project start time and the project size. The analysis results suggested there are some risk factors which have relatively strong and stable

relationships to cost overrun. Project managers should care the followings especially, to avoid cost overrun.

- Reviews with the customer and approval.
- The quantitative estimation by the tool.
- The estimation of the systemization size and the basis for it.
- The clarity of customer's project goals.
- The clarity of deliverables and products for each task.

Based on the above analysis, we selected some risk factors, and used them as independent variables of cost overrun prediction. We applied collaborative filtering to predict cost overrun, and experimental result showed that cost overrun projects can be predicted using the risk factors. Our future work is identifying other risk factors which are more effective to predict cost overrun projects.

ACKNOWLEDGMENT

This work is being conducted as a part of the StagE project, The Development of Next-Generation IT Infrastructure, and Grant-in-aid for Young Scientists (B), 22700034, 2010, supported by the Ministry of Education, Culture, Sports, Science and Technology. We are deeply grateful to people who cooperated for data collection. Also, we would like to thank Dr. Naoki Ohsugi for offering the collaborative filtering tool.

REFERENCES

[1] J. Hanley, and B. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," Radiology, no.143, pp.29-36, 1982.

[2] B. Kitchenham, S. Pfleeger, B. McColl, and S. Eagan, "An empirical study of maintenance and development estimation accuracy," Journal of Systems and Software, vol.64, no.1, pp. 57-77, 2002.

[3] S. Lessmann, B. Baesens, C. Mues, and S. Pietsch, "Benchmarking Classification Models for Software Defect Prediction: A Proposed Framework and Novel Findings," IEEE Transactions on Software Engineering, vol.34, no.4, pp.485-496, 2008.

[4] J. Procaccino, J. Verner, S. Overmyer, and M. Darter, "Case study: factors for early prediction of software development success," Information and Software Technology, vol.44, no.1, pp.53-62, 2002.

[5] Project Management Institute, A Guide to the Project Management Body of Knowledge: (PMBOK Guide). p.459, Project Management Institute, 2008.

[6] Software Engineering Center, Information-Technology Promotion Agency, Japan, "MIERUKA (Visualization)" of IT Projects: Upstream Process, Software Engineering Center, Information-Technology Promotion Agency, Japan, 2010. http://www.ipa.go.jp/english/sec/reports/20100507b/20100507b_Upsteam.pdf

[7] Software Engineering Center, Information-Technology Promotion Agency, Japan, "MIERUKA (Visualization)" of IT Projects: Midstream Process, Nikkei Business Publications, 2008 (in Japanese).

[8] Software Engineering Center, Information-Technology Promotion Agency, Japan, "MIERUKA (Visualization)" of IT Projects: Downstream Process, Software Engineering Center, Information-Technology Promotion Agency, Japan, 2010. http://www.ipa.go.jp/english/sec/reports/20100507/20100507b_Downsteam.pdf

[9] M. Shepperd, and C. Schofield, "Estimating software project effort using analogies," IEEE Transactions on Software Engineering, vol.23, no.12, pp.736-743, 1997.

[10] Y. Takagi, O. Mizuno, and T. Kikuno, "An Empirical Approach to Characterizing Risky Software Projects Based on Logistic Regression Analysis," Empirical Software Engineering, vol.10, no.4, pp.495-515, 2005.

[11] M. Tsunoda, A. Monden, and K. Matsumoto, "Sharpe Ratio Based Index for Building Fault Prediction Model," Supplemental Proc. International Symposium on Software Reliability Engineering (ISSRE 2011), Vol.1, No.6, pp.1-2, 2011.

[12] M. Tsunoda, A. Monden, J. Shibata, and K. Matsumoto, "Empirical Evaluation of Cost Overrun Prediction with Imbalance Data," Proc. International Conference on Computer and Information Science (ICIS 2011), pp.415-420, 2011.

[13] J. Verner, W. Evanco, and N. Cerpa, "State of the practice: An exploratory analysis of schedule estimation and software project success prediction," Information and Software Technology, vol.49, no.2, pp.181-193, 2007.