

# Characterization of Risky Projects based on Project Managers' Evaluation

**Osamu Mizuno, Tohru Kikuno**  
Graduate School of Engineering Science,  
Osaka University  
Toyonaka-City, Osaka, 560-8531, Japan  
{o-mizuno, kikuno}@ics.es.osaka-u.ac.jp

**Yasunari Takagi, Keishi Sakamoto**  
OMRON Corporation  
Kusatsu-City, Shiga, 525-0035, Japan  
{taka, sakamoto}@biwa.kusatsu.omron.co.jp

## ABSTRACT

During the process of software development, senior managers often find indications that projects are risky and take appropriate actions to recover them from this dangerous status. If senior managers fail to detect such risks, it is possible that such projects may collapse completely.

In this paper, we propose a new scheme for the characterization of risky projects based on an evaluation by the project manager. In order to acquire the relevant data to make such an assessment, we first designed a questionnaire from five viewpoints within the projects: requirements, estimations, team organization, planning capability and project management activities. Each of these viewpoints consisted of a number of concrete questions. We then analyzed the responses to the questionnaires as provided by project managers by applying a logistic regression analysis. That is, we determined the coefficients of the logistic model from a set of the questionnaire responses. The experimental results using actual project data in Company A showed that 27 projects out of 32 were predicted correctly. Thus we would expect that the proposed characterizing scheme is the first step toward predicting which projects are risky at an early phase of the development.

## Keywords

Software risk management, Questionnaire, Risky project, Logistic regression

## 1 INTRODUCTION

Since 1993 we have been involved in software process improvement[6] at a certain company (let's call it Company A for the sake of convenience). At Company A, the Software Engineering Process Group(SEPG) has been in existence for seven years, and has tried to extend process improvement into their company. We have already succeeded in achieving several actual improvements[9, 10, 13, 14]. This study is also a part of the SEPG's activities in 1999 in the field of software

development at Company A.

Previous studies focused mainly on the quality of the final products by comparing projects before and after improvement. Most of these comparisons, however, were carried out after the development activities had already ended. The SEPG really needs to observe the development itself in order accurately to perceive process improvement. In fact, when each project was looked at more closely, it was found that a number of projects appeared to be out of control from the senior managers' viewpoint and either recovered by themselves or were brought back under control by the senior managers' leadership. A few really did collapse, and although the number of such cases was quite small, this should clearly be avoided as much as possible.

Projects in such an uncontrolled state have also been seen in other organizations[3]. Hereafter we shall refer to such projects as "risky projects." In order to bring risky projects back under control, it is necessary first to identify which projects are risky and then understand the signs which indicate that the project is risky.

The idea of risk management has been suggested as one way of dealing with risky projects[3, 4, 8, 12]. Risk management aims to identify the risk factors which may be the reason for loss of control, and mitigate the risks in the projects. However, since there are so many risk factors in software development, it is difficult to check all possible risk factors in an actual project. It follows that there is a high expectation that a reasonable range of risk factors may be investigated in the development field.

On the other hand, Yourdon proposed the notion of the *death march project*[15]. The definition of 'death march project' mainly focuses on unreasonable customer requirements, which cannot be met by the developers themselves. Fortunately, there are very few projects at Company A which suffer from 'the death march project' syndrome. Nevertheless, efforts should be made to try to identify the projects which may become 'death march projects', and senior management should be informed of the difficulties.

In order to characterize risky projects, we have experimented with investigations using the data sets from previous projects. We tried, in fact, to show a correlation between software

metrics[5] and risky projects, but without success. We then changed our approach and tried to acquire technical knowledge or knowhow from project managers or senior managers. On a basis of 7 years experience, the SEPG considers that project managers or senior managers are able clearly to recognize risky projects. We therefore designed the questionnaire to bring out their intuitive knowledge regarding risky projects or their recognition of risks in development. The questionnaires were then passed on project managers, whose cooperation in filling in the facts was requested, and the responses were collected. We then statistically analyzed the responses and thus succeeded in identifying risky projects. This paper describes: (1) the design of the questionnaire; and (2) the statistical analysis of the responses.

In designing the questionnaire, we took into account a number of references[3, 4, 8, 12] regarding risk management, and such manuals on analyzing risky projects as are currently used at Company A. We then summarized the factors defining risky projects and classified them according to the following five areas: (1) Requirements, (2) Estimations, (3) Team organization, (4) Planning capability, and (5) Project management activities.

We then used the logistic regression analysis for the statistical analysis of the responses from project managers. Logistic regression is a standard classification technique based on maximum likelihood estimation. Since this paper considers projects at Company A regarding which the SEPG is able to distinguish risky projects from others, the logistic regression model is suitable for this type of analysis. In the actual study we divided the entire data set into two according to the periods when they were carried out: the data for projects from 1996 to 1997 were used to construct a logistic model, while the data for projects in 1998 were applied to achieve an effectiveness analysis for the constructed model. We then constructed a logistic model using the former data set. In the case of the constructed model, statistical significances and the goodness-of-fit are indicated. Effectiveness was subsequently analyzed in order to predict risky projects using the latter data set. The results of this analysis sufficiently confirmed the validity of our approach.

The rest of this paper is organized as follows: Section 2 defines the notion of a risky project. Section 3 shows the outline of our approach to the identification and prediction of risky projects. The design of the questionnaire intended to harvest the knowledge of project managers is described in Section 4. The application of the proposed approach to real project data is shown in Section 5. Section 6 shows an experimental evaluation of the effectiveness of the proposed method. Finally, Section 7 summarizes this paper and discusses future work.

## 2 WHAT DO RISKY PROJECTS IMPLY?

### The SEPG's activities at Company A

The main products of Company A consist of embedded software in ticket vending machines, ATMs(automated teller

machines) and POS(point of sales) systems. The software process adopted in Company A is the waterfall model. To promote process improvement in Company A, the SEPG was established 7 years ago. Various activities undertaken by the SEPG, such as the collection of software metrics and the formalization of review activity, have been carried out with an eye to process improvement. Software metrics and project data such as productivity, quality and duration are therefore constantly collected from all the software development projects within Company A. The SEPG both recognizes and reports that the average values of both quality and productivity for all projects at Company A have increased steadily year by year.

However, the SEPG also observed the fact that not a few projects tend to be in a somewhat uncontrolled state for a certain period of their development, and that most of them return to a controlled state by themselves or by virtue of the senior managers' leadership. A certain proportion, however, actually collapses. While the number of such cases is quite small, they should clearly be avoided as much as possible.

### Intuitive explanation of risky projects

Here we give an intuitive definition of *risky projects*. It is well known that in actual software development a project seldom reaches a situation where the developers cannot control the software development, or need considerable help from a senior manager in order to get the project back on track. In this paper, we refer to such projects as *risky projects*. A similar type of a project has been dubbed the *death march project*[15]. The 'death march project' describes a project which does not have sufficient project resources and thus eventually fails in the course of its development. However, as mentioned before, most risky projects to be targeted in this paper are able to be brought back under control state and finally deliver the products successfully. It is possible, however, for them to collapse temporarily at the same stage during the software development period.

### Risky projects determined by the SEPG

As shown in Figure 1, the SEPG has until now determined whether a project is risky or not on a basis of the final re-

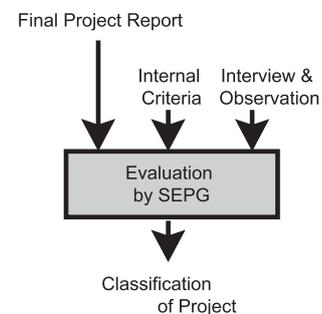


Figure 1: Legacy evaluation process of risky projects

port of the project (thus after the development activities of the project are finished). To determine risky projects, the SEPG uses two quantitative criteria with respect to cost and duration: the difference between actual cost and estimated cost, and the difference between actual duration and estimated duration. That is, risky projects exceeded cost estimates by  $\alpha\%$ , and overran time estimates by  $\beta$  months<sup>1</sup>. These internal criteria regarding cost and duration are what the SEPG uses to identify and define risky projects.

However, these criteria alone are not sufficient to determine risky projects, as there are some projects that should be defined as risky according to the above rule, but actually causes no problems. The SEPG therefore investigates each project in greater detail on a basis of observations by the SEPG and interviews with the project managers and the developers. Lastly, risky projects are defined on a basis of investigations carried out by the SEPG. Although the decision process partially includes subjective evaluation by the SEPG, it has nevertheless been accepted by the whole division at Company A.

### 3 OUTLINE OF OUR APPROACH

As mentioned before, the SEPG considers that project managers or senior managers can recognize the sign of risky projects consciously, but they cannot say or hesitate to say that the project is out of control. The reason for their hesitation may be a lack of the basic recognition, pressure to adhere to a schedule, and so on. So, in this paper, we aim to clarify the impact of each risk factor and construct a basis for identifying risky projects statistically.

Figure 2 shows the outline of our approach for identifying risky projects. First, in Step 1, we designed a questionnaire to be distributed to project managers in order to collect the risk assessment data. The questionnaire consists of five viewpoints of a development, each being further divided into several risk factors for risky projects. Next, in Step 2, the SEPG distributed the questionnaire to concerned project managers, and asked them to fill in the questionnaire. After they finished filling in the questionnaire, the SEPG collected them. Then, in Step 3, from the responses to the questionnaire, we obtained the risk assessment data, and then constructed a logistic model to predict risky projects.

Put simply, the questionnaire is a list of risk factors, and it is classified into five viewpoints of the software development: requirements, estimations, team organization, planning capability and project management activities. (The details of the design will be described in Section 4.) Each sub-item regarding risk factors in the questionnaire must be filled in according to the Likert scale[5]: “Strongly Agree”, “Agree”, “Neither agree nor disagree” or “Disagree”.

Logistic regression, a standard classification technique in the experimental sciences, has already been used in software

<sup>1</sup> Due to the contract with Company A, the concrete values of criteria  $\alpha$  and  $\beta$  cannot be published here.

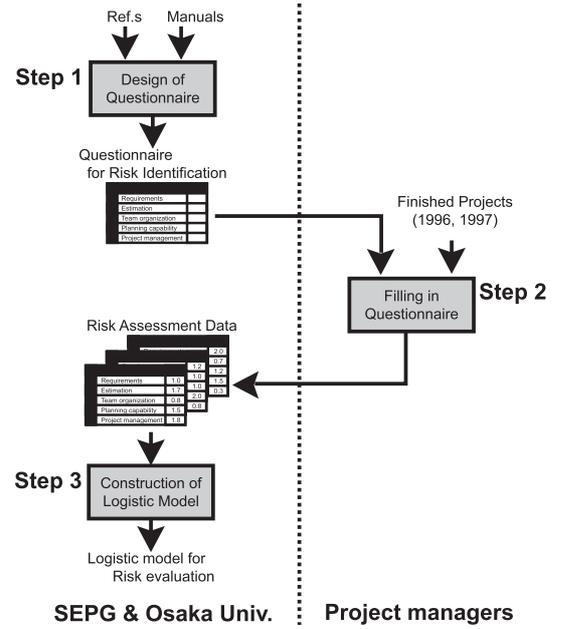


Figure 2: Outline of characterization of risky projects

engineering to predict error-prone components[1, 2, 11].

A logistic model is based on the following equation:

$$E(Y|x_1, \dots, x_n) = \frac{e^{b_0 + b_1 x_1 + \dots + b_n x_n}}{1 + e^{b_0 + b_1 x_1 + \dots + b_n x_n}}$$

where  $x_1, \dots, x_n$  are explanatory variables in the model, and  $Y$  is a binary dependent variable which represents a project is risky or not.  $E$  is the conditional probability that  $Y = 1$  (i.e. a project is risky) when the values of  $x_1, \dots, x_n$  are determined. We select the five viewpoints in the questionnaire as potential explanatory variables, and estimate the coefficients  $b_i$ 's using the risk assessment data obtained from the responses to the questionnaire.

### 4 QUESTIONNAIRE DESIGN

#### Five viewpoints

In this study, we have investigated various references [3, 4, 8, 12] regarding the risk management and the manuals used at Company A for analyzing risky projects. Based on the results of this investigation, we have summarized all key risk factors and classified them into the following five viewpoints: (1) Requirements, (2) Estimations, (3) Team organization, (4) Planning capability, and (5) Project management activities.

As mentioned above, we derived some risk factors from the manual now being extensively used at Company A. The first version of the manual was laid out 7 years ago, and it has been updated or revised to reflect the knowledge acquired from the experiences of development.

The overview of the questionnaire is shown in Figure 3. The project managers responded to the questionnaire by filling

Items		Evaluation
<b>1. Requirements</b>		
1.1	Unreasonable customers.	
1.2	Developers could not elucidate sufficient requirements.	
1.3	Developers misunderstood the requirements of the customer.	
1.4	Lack of interactive agreement regarding requirement specifications between the customer and the developer.	
<b>2. Estimations</b>		
2.1	There were missing items to be estimated; these items were included in the implicit requirements.	
2.2	The importance of estimations was not well recognized.	
2.3	Non-technical pressure rendered estimates of costs and/or schedules unrealistic.	
2.4	Over-optimism in estimating technical issues.	
2.5	Insufficient estimations were carried out using the results of successful projects in the past.	
<b>3. Team Organization</b>		
3.1	Wrong people available (lack of skills, lack of training, lack of expertise).	
3.2	Incorrect staffing (too few people for current task).	
<b>4. Planning Capability</b>		
4.1	Unclear responsibilities and authorities.	
4.2	Inadequate specifications regarding the work product.	
4.3	Inadequate or excessive planning or scheduling of the review process.	
4.4	Lack of commitment on the part of all of the developers with regard to the project plan.	
4.5	Lack of review for the project plan by senior managers.	
4.6	Inadequate control of the development process.	
<b>5. Project Management Activities</b>		
5.1	Lack of risk management on technical matters.	
5.2	Low morale on the part of the developers.	
5.3	Lack of perception on the part of the managers to ensure a concerned effort.	
5.4	Requirement or specification changes were not managed sufficiently.	
5.5	Lack of progress reporting.	
5.6	Lack of data needed to keep track of a project.	

Figure 3: Problem Analysis Questionnaire

in each sub-item with “Strongly Agree”, “Agree”, “Neither Agree nor Disagree” or “Disagree” according to the Likert scale. In the following subsections, we explain the details of each viewpoint and risk factor.

### (1) Requirements

The *Requirements* viewpoint includes risk factors which are related to the requirement definitions and the translation of the requirements into specifications.

It is well known that excessive, immature, unrealistic, or unstable requirements on the part of customers drastically affect the success of a project. In most cases the problems related to requirements suddenly appear at the end of a project. To make matters worse, most of them have an effect on the fundamental implementation of a product. Projects with problems in requirements therefore tend to be risky.

From the viewpoint of the requirements, the risk factors are distinguished as follows:

#### (1.1) Unreasonable customers.

This item checks whether the requirements as stated by the customer are clear and consistent. It is important for the developers to understand that what the customer wants to achieve is not clear, or that the requirements as stated by the customer may not be consistent.

#### (1.2) Developers could not elucidate sufficient requirements.

This item checks whether the developers have elucidated sufficient requirements to produce the system and/or software. Customers do not always have sufficient technical knowledge regarding their needs. Developers should try to elucidate sufficient requirements, which may include implicit ones, from the customers.

#### (1.3) Developers misunderstood the requirements of the customer.

This item checks whether the developers have sufficient skills to understand the requirement. The developers should have not only sufficient technical knowledge for the project, but also command specific knowledge regarding the customers’ areas of application.

#### (1.4) Lack of interactive agreement regarding requirement specifications between the customer and the developer.

This item checks whether a review of documents has been carried out by both the developers and the customer. In order to confirm that the developers understand the customer’s requirements properly, it is very important for both of them to have meetings to review the requirement.

### (2) Estimations

The *Estimations* viewpoint includes risk factors related to the estimations and the technical methods for carrying out the estimations.

In order to construct a project plan, the project size should be estimated as exactly as possible. Furthermore, functionalities to be implemented must be calculated approximately. If optimistic or insufficient estimations are carried out, then several problems inevitably arise in a project.

From the estimations viewpoint, the risk factors are distinguished as follows:

#### (2.1) There were missing items to be estimated; these items were included in the implicit requirements.

This item checks whether there are implicit requirements which are currently not included in the estimation. Usu-

ally, in the development of embedded systems, most of the implicit requirements are related to failure recovery. Roughly speaking, since the size of components for failure recovery tends to become large, underestimation might cause serious confusion in development.

- (2.2) The importance of estimations was not well recognized.

This item checks whether the importance of estimations is well recognized by the developers. No estimations means that the objective of the project is not clear, and thus appropriate or adequate management cannot be expected for the project.

- (2.3) Non-technical pressure rendered estimates of costs and/or schedules unrealistic.

This item checks that the developers do not easily yield to any non-technical pressure (for example, unreasonable reduction in costs or schedule). Generally speaking, developers tend to give way if they do not have concrete reasons for their estimations. As a result, they find themselves obliged to be involved in so called "death march projects".

- (2.4) Over-optimism in estimating the technical issues.

This item checks whether the developers have estimated technical issues with which they are not very familiar in an over-optimistic way. Usually, over-optimism regarding the estimations tends to cause underestimation. For example, the developers may not consider the need to prepare activities dealing with risks in the development.

- (2.5) Insufficient estimations were carried out using the result of successful projects in the past.

This item checks whether the developers are estimating the current project by simply referring to similar projects from the past, which they may have developed themselves. Naturally it is advisable with regard to estimations which have provided positive experience from similar projects in the past be put to good use. But at the same time, it is very dangerous to depend on them too much with inadequate analysis.

### (3) Team organization

The *Team organization* viewpoint includes risk factors which are related to the staffing of the projects and the fundamental skills of the developers of the projects.

Stable organization is essential to ensure that a project be carried through successfully. It is well known that unstable organization can be an essential reason for risky projects (especially for large projects). It is also very important for the success of a project that the developers have the correct skills. But problems caused by the lack of the developers' skills are generally hard to find in the early phases of projects.

From the viewpoint of team organization, the risk factors are distinguished as follows:

- (3.1) Wrong people available (lack of skills, lack of train-

ing, lack of expertise).

This item checks whether the skill level needed for the development has been clarified and whether a sufficient number of developers with sufficient skill levels can be mustered. Needless to say, it is important to clarify the skill level needed for the development. However in some cases, such important tasks could be forgotten or neglected due to limitations of human resources.

- (3.2) Incorrect staffing (too few people for the current task).

This item checks whether the project is well staffed. At the beginning, project managers tend to think optimistically. Thus they may not take a lack of human resources seriously. They consider or expect that a project will be managed quite well if the available developers bring their abilities into full play.

### (4) Planning capability

The *Planning capability* viewpoint includes risk factors which are related to the planning or scheduling method and the resulting project plan.

For the purpose of the initial requirements, the project is planned by clarifying responsibilities, preparing documentations for the products and making a concrete schedule for the project. An inadequate project plan clearly disturbs the smooth flow of the execution of the project. Since the project plan is a kind of promise regarding the way the project will be carried out, all of the developers should understand the project plan as presented.

From the viewpoint of planning capability, the risk factors are distinguished as follows:

- (4.1) Unclear responsibilities and authorities.

This item checks that the project has been systematically divided into technical activities by using the Work Breakdown Structure (WBS), and whether the responsibility for each technical activity has also been clearly specified. Unclear responsibility may prevent the early detection of serious problems concealed in the project.

- (4.2) Inadequate specifications regarding the work product.

This item checks that work products to be produced by the development have been correctly specified. If the work products themselves have not been specified exactly in the project plan, then the responsibility is often not clear for each product.

- (4.3) Inadequate or excessive planning or scheduling of the review process.

This item checks whether the review process plan or the stages in the schedule have been adequately specified in the project plan. The review processes mostly tend to be skipped or shortened because of pressure from the schedule. So the period and the objectives of the review should be included in the project plan.

- (4.4) Lack of commitment on the part of all of the devel-

opers with regard to the project plan.

This item checks whether the plan has been reviewed by all of the developers and managers of the project. All engineers engaged on the project should recognize the project plan, and understand the concrete objective of the project.

- (4.5) **Lack of review for the project plan by senior managers.**

This item checks whether the senior management has reviewed the project plan or not. The fact that senior management has actually reviewed the project plan means that the project is officially identified within the company which has a very positive impact on the success of the project.

- (4.6) **Inadequate control of development process.**

This item checks whether the methodology for controlling or managing the project has been specified in the project plan. Delays in the schedule or any problems in the development process should be reported immediately to senior management. The system or framework for reporting problems therefore be established and specified in the project plan clearly.

### **(5) Project management activities**

The *Project management activities* viewpoint includes risk factors which are related to project management activities during development.

Even if a good project plan is constructed, the project can be risky when the managerial activities based on the project plan are not undertaken effectively and opportunely.

From the project management activities viewpoint, the risk factors are distinguished as follows:

- (5.1) **Lack of risk management on technical matters.**

This item checks whether the project is well managed technically in the sense that the occurrence of a technical problem is immediately detected. Usually such technical problems need special skills to solve. It is usually very difficult to assign a special engineer with specific technical skills the moment after such a technical problem occurs in the project.

- (5.2) **Low morale on the part of the developers.**

This item checks whether the morale of the developers is low or not. For instance, if the developer feels that someone else can deal with the delay of his/her own activity, then the developer's morale is clearly low. Low morale usually makes it difficult to detect even small problems which eventually may become very serious and cause huge delays in a project.

- (5.3) **Lack of perception on the part of the managers to ensure a concerted effort.**

This item checks that the developers are actually working on the assigned project. The developer may frequently be disturbed by problems deriving from the previous project on which he was engaged. The project

manager should carefully manage developers so that they are not excessively distracted by problems from previous projects.

- (5.4) **Requirement or specification changes were not managed sufficiently.**

This item checks that the changes in requirements or specifications are appropriately managed and the developers kept informed. It is well known that uncertain changes in requirements or specifications confuse the developers and finally result in a risky project.

- (5.5) **Lack of progress reporting.**

This item checks whether the developers are providing the project managers with regular progress reports. In the developmental environment where a project manager criticizes an erring developer bad timing in reporting problems on the project may cause delays. In the worst case no reporting is done at all.

- (5.6) **Lack of data needed to keep track of a project.**

This item checks whether the project managers are able objectively to keep track of a project on a basis of the software metrics which were collected during development. If such data are not available, then correct recognition of the project, and thus appropriate management decisions, will be difficult.

## **5 ANALYSIS BY LOGISTIC REGRESSION**

In this section, we apply the proposed questionnaire in Section 4 to sample projects, and successively apply logistic regression analysis to the risk assessment data obtained from the questionnaire.

### **Example projects**

We chose 32 projects which were undertaken from 1996 to 1997 by Company A. These were projects concerned with developing embedded software for ticket vending machines, ATMs(automated teller machines) and POS(point of sales) systems. As mentioned before, the development process for these projects is the overlapping waterfall model[7].

Note that these 32 projects did not represent the entirety of projects performed by Company A. They were specifically those projects which were carefully watched or monitored by the SEPG because of their characteristics(for example, the previous project by the same team had failed, the estimated size of the project was larger than a certain degree, and so on).

Since all of these projects completed their development, the SEPG had already identified risky projects according to the decision process mentioned in Section 2. As a result, 10 projects out of 32 were classified as risky projects(Again, note that this never means that 1/3 of Company A's projects are risky.). Thus Table 1 has a column "Final status", which shows the result of classification by the SEPG.

### **Result of Step 2(Risk assessment data)**

We distributed the questionnaires designed in Section 4 to the

Table 1: Example projects in '96 and '97

Projects in '96 and '97	Requirements	Estimations	Team Organization	Planning Capability	Project Management Activity	Final status
PJ1	1.50	1.00	0.50	0.83	1.00	Risky
PJ2	1.50	1.00	0.50	1.17	1.00	Risky
PJ3	0.75	1.40	1.00	0.50	0.17	Risky
PJ4	1.50	0.40	1.00	0.83	0.83	Risky
PJ5	1.00	1.00	0.50	0.50	0.83	Risky
PJ6	0.75	0.00	0.00	1.00	0.33	Risky
PJ7	1.50	1.40	2.00	1.67	1.17	Risky
PJ8	1.75	1.80	2.00	1.83	1.33	Risky
PJ9	0.75	0.60	0.00	1.00	0.83	Risky
PJ10	0.00	0.00	2.00	2.00	1.00	Risky
PJ11	1.75	1.20	1.50	0.67	1.00	No problem
PJ12	0.50	0.00	0.00	0.00	0.00	No problem
PJ13	1.00	0.00	0.50	0.67	1.17	No problem
PJ14	0.00	0.00	0.00	0.00	0.33	No problem
PJ15	0.00	0.00	0.00	0.33	0.17	No problem
PJ16	0.00	0.20	0.00	1.17	0.33	No problem
PJ17	0.00	0.80	1.50	1.17	0.67	No problem
PJ18	0.75	0.40	1.50	0.00	0.67	No problem
PJ19	0.00	0.00	0.00	0.00	0.50	No problem
PJ20	0.50	0.00	1.00	1.17	0.50	No problem
PJ21	0.75	0.20	0.00	0.33	0.17	No problem
PJ22	1.25	0.80	0.00	0.00	0.50	No problem
PJ23	0.00	0.40	0.00	0.33	0.50	No problem
PJ24	0.50	0.00	0.00	0.67	0.50	No problem
PJ25	1.25	0.40	0.00	0.67	0.83	No problem
PJ26	0.75	0.00	0.00	0.50	0.17	No problem
PJ27	0.75	0.80	0.00	0.17	0.17	No problem
PJ28	0.00	0.00	0.00	0.50	0.33	No problem
PJ29	1.50	0.40	0.50	0.50	0.50	No problem
PJ30	0.00	0.60	0.00	0.17	0.33	No problem
PJ31	0.00	0.00	0.00	0.17	0.00	No problem
PJ32	0.00	1.20	0.50	0.17	0.33	No problem

project managers or the person responsible for the 32 target projects, and explained the details of questionnaire and the purpose of the trial. The responses to the questionnaire were collected by the SEPG after a month.

In order to elucidate the risk assessment data from the responses, we assigned points 2, 1 and 0 to “Strongly Agree”, “Agree” and “Disagree” for each evaluation(in Figure 3), respectively. We translated the evaluation “Neither agree nor disagree” into “Agree”, because it was guessed that project managers had some problem that was not quite serious enough to select “Disagree”.

For further analysis in Step 3, we summed up the points of the sub-questions. Then we standardized them for each viewpoint by the number of small questions in it. The points for each viewpoint thus ranges from 0 to 2. The values of the five viewpoints for each project are shown in Table 1. We shall hereafter refer to them as the risk assessment data. As mentioned above, the result of the observation by the SEPG (that is, final status) for each project is also shown in Table 1.

**Result of Step 3 (Construction of logistic model)**

We then considered the five viewpoints in the questionnaire as the parameters of a logistic model. The correlation coefficients between the five parameters are shown in Table 2. Some of them show relatively high correlation.

From the risk assessment data shown in Table 1, we constructed a logistic model. Two parameters were adopted in the logistic model using the stepwise method: *estimations*( $x_1$ ) and *planning capability*( $x_2$ ). The coefficients for both of them(that is,  $b_1$  and  $b_2$ , respectively) were determined as

Table 2: Correlation coefficients among parameters in the model

	Estimation	Team Organization	Planning Capability	Project Management Activity
Requirements	0.547	0.363	0.295	0.608
Estimation	-	0.521	0.283	0.472
Team Organization		-	0.649	0.645
Planning Capability			-	0.653

Table 3: Coefficients in the logistic model

	Coefficient	odds ratio	p-value
Intercept	-5.251 ( $b_0$ )	0.005	0.0041
Estimation ( $x_1$ )	2.727 ( $b_1$ )	15.288	0.0005
Planning Capability ( $x_2$ )	3.984 ( $b_2$ )	53.749	0.0175

shown in Table 3. The  $p$ -value in Table 3 represents statistical significance. It represents the probability that the coefficient being different from zero by chance. A significance threshold of 0.05 has often been used. Here, we used a likelihood ratio test with hypotheses  $H_0 : b_1 = 0$  and  $H_0 : b_2 = 0$ , and both values of the coefficients were shown to be statistically significant. Furthermore, the correlation coefficient between two parameters *estimations* and *planning capability* is 0.283(from Table 2). We can say that these two parameters are statistically independent, and it is valid to include these two parameters in the model.

“Odds ratio” in Table 3 represents the ratio between the probability of a project being a “risky project” and the probability of it being a “no problem” project(that is, not risky) when the value of the explanatory variable increases by one unit. Intuitively speaking, it represents the impact of the explanatory variable in causing projects to be risky.

We then investigated the significance and goodness-of-fit of the whole model. First, the hypothesis  $H_0 : b_1 = b_2 = 0$  was tested by the likelihood ratio test. Table 4 showed that the  $p$ -value of the model was less than 0.0001. This model was thus shown to be statistically significant. Now, deviance is an indicator used to check whether the model fitted the original data. As the value of  $\chi^2$  in Table 4 is close to the degree of freedom, it can be said that the model fits closely. Here, the value of  $\chi^2$  is 20.596 and the degree of freedom is 25. So, we can say that the fit is good.

Table 5 shows the result of prediction by the constructed

Table 4: Statistics for the model

	Degree of freedom	$\chi^2$	p-value
Deviance	25	20.596	0.7149
Test of likelihood ratio	2	19.154	< 0.0001

Table 5: Results of prediction in '96 and '97

Actual	Predicted	
	No problem	Risky
No problem	18	4
Risky	1	9

Table 6: Computed probabilities

Projects in 1998	Estimation ( $x_1$ )	Planning Capability ( $x_2$ )	$E(Y x_1, x_2)$	Final Status
PJ33	0.60	2.00	0.987	Risky
PJ34	1.20	1.33	0.966	Risky
PJ35	1.20	0.33	0.343	Risky
PJ36	0.00	0.17	0.010	No problem
PJ37	0.40	0.33	0.056	No problem
PJ38	0.20	0.33	0.033	No problem
PJ39	0.00	0.17	0.010	No problem
PJ40	0.40	0.67	0.182	No problem

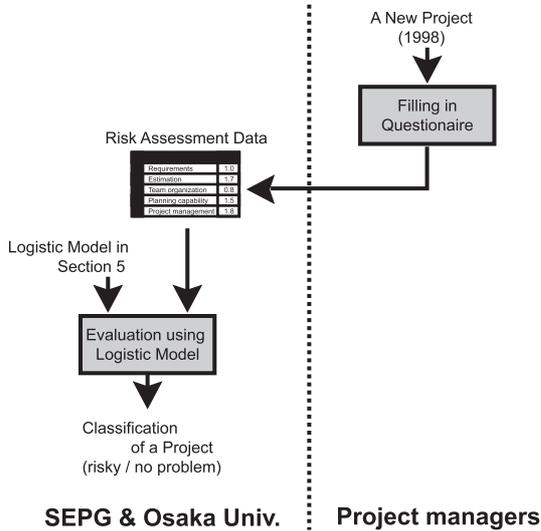


Figure 4: Outline of experimental prediction

model for the risk assessment data set from 1996 to 1997. The projects in Table 5 were divided into two groups: risky or not. Here, we assume that the classification threshold is 0.3.<sup>2</sup> It was shown that 27 projects out of 32 were predicted correctly. Most important was that 9 out of ten actual risky projects (they were seen to be “risky” in Table 1) were predicted correctly.

## 6 APPLICATION TO NEW PROJECTS

### Outline

In order to show the effectiveness of the proposed characterization, we performed an experimental prediction which applies the proposed framework to the risk assessment data set of the 1998 projects (Please note that these 1998 projects are not included in the data set used in Section 5.).

The outline for predicting risky projects is shown in Figure 4. In the prediction of risky projects, the questionnaires were again distributed to managers of new projects. Based on the collected responses to the questionnaire and the logistic model constructed in Section 5, we computed the probability of the projects being risky.

<sup>2</sup>We don't at present have any confidence in this classification threshold. But in general, the prediction of risky projects should cover as many risky projects as possible. That is the reason why we selected a threshold below 0.5.

Table 7: Result of prediction in '98

Actual	Predicted	
	No problem	Risky
No problem	5	0
Risky	0	3

### Experimental prediction

There were 8 projects in 1998 suitable for this experiment (Note again that these projects did not represent the entirety of the projects in 1998.). As in Section 5, we collected the responses to the questionnaire and worked out the risk assessment data set for the logistic regression analysis.

We then applied the logistic model constructed in Section 5 to the risk assessment data set for the projects in 1998, and calculated the probability of their being risky. The results are summarized in Table 6.

In Table 6,  $E(Y|x_1, x_2)$  shows the computed probability of a project being risky. We again assume that the threshold of the probability of a project being risky is 0.3. We can see the probabilities (that is,  $E(Y|x_1, x_2)$ 's) of PJ33, PJ34 and PJ35 are higher than 0.3. That is, these three projects were predicted to be risky projects. The final status of these projects is also shown in Table 6. In Table 7, comparing the probabilities with the final status, we can see that all projects were predicted correctly.

### Discussion

We do not insist that the explanatory variables *estimations* and *planning capability* are useful in general for predicting risky projects. Our objective for this study was to propose a new approach to the identification prediction of risky projects using the questionnaire and the logistic model.

In this experimental prediction, the questionnaire was distributed to the managers of finished projects. In such a case, the managers tended to respond quite honestly. We were thus able to obtain a rather idealized result. In running real projects, however, it may be doubtful that all of the project managers would respond honestly. We must make put more thought into our method of collecting the responses to the questionnaires when we distribute them at an early phase of

a development project (such as the end of the module design phase or the coding phase at the latest).

At this point, the questionnaire may include questions which cannot be responded at an early phase of the development. However, as for the *estimations* and *planning capability* viewpoints, we can say that most sub-questions in them can be responded even at an early phase. We therefore think that the proposed approach is easily applicable to the field of real development albeit with some modification. We are therefore now planning to apply it to live projects at Company A.

## 7 CONCLUSION

In this paper, we have proposed a new approach to the identification of risky projects which might cause disastrous problems during development. In order to determine key risk factors in risky projects, we designed a questionnaire to be distributed to project managers of the real projects from 1996 to 1997. Based on the responses to the questionnaire, we collected a risk assessment data set and applied logistic regression analysis in order to obtain a logistic model. The model was shown to be statistically significant, and the goodness-of-fit of the model was also good. We also carried out an effectiveness analysis of the constructed model. The result showed that the constructed model can nicely predict risky projects in the new data set in 1998. Future research includes the prediction of risky projects in live software development projects.

## ACKNOWLEDGEMENTS

Authors would like to thank Associate Prof. Shinji Kusumoto of Osaka University for his discussions and advice to our analysis in this paper. We also would like to thank anonymous reviewers for their useful comments.

## REFERENCES

- [1] V. R. Basili, L. C. Briand and W. L. Melo, "A validation of object-oriented metrics as quality indicators," *IEEE Trans. on Software Eng.*, vol. 22, no. 10, pp. 751–761, 1996.
- [2] L. C. Briand, V. R. Basili and C. Hetmanski, "Developing interpretable models with optimized set reduction for identifying high risk software components," *IEEE Trans. Software Eng.*, vol. 19, no. 11, pp. 1028–1044, 1993.
- [3] E. H. Conrow and P. S. Shishido, "Implementing risk management on software intensive projects," *IEEE Software*, Vol. 14, No. 3, pp. 83–89, 1997.
- [4] R. Fairley and P. Rook, "Risk management for software development," In *Software Engineering*, IEEE CS Press, pp. 387–400, 1997.
- [5] N. E. Fenton and S. L. Pfleeger, *Software Metrics : A Rigorous & Practical Approach*, PWS Publishing, 1997.
- [6] W. S. Humphrey, *Managing the Software Process*, Addison Wesley, Reading, MA, 1989.
- [7] W. S. Humphrey, *A Discipline for Software Engineering*, Addison Wesley, Reading, MA, 1995.
- [8] D. W. Karolak, *Software Engineering Risk Management*, IEEE CS Press, CA, 1996.
- [9] O. Mizuno, T. Kikuno, K. Inagaki, Y. Takagi and K. Sakamoto, "Analyzing effects of cost estimation accuracy on quality and productivity," In Proc. 20th International Conference on Software Engineering (ICSE'98), pp. 410–419, 1998.
- [10] O. Mizuno and T. Kikuno, "Empirical evaluation of review process improvement activities with respect to post-release failure," In Summary of Empirical Studies on Software Development Engineering (ICSE'99 Workshop), pp. 50–53, 1999.
- [11] J. Munson and T. Khoshgoftaar, "The detection of fault-prone programs," *IEEE Trans. Software Eng.*, vol. 18, no. 5, 1992.
- [12] F. J. Sisti and S. Joseph, "Software risk evaluation method version 1.0," Technical Report CMU/SEI-94-TR-19, Software Engineering Institute, 1994.
- [13] Y. Takagi, T. Tanaka, N. Niihara, K. Sakamoto, S. Kusumoto and T. Kikuno, "Analysis of review's effectiveness based on software metrics," In Proc. 6th International Symposium on Software Reliability Engineering (ISSRE'95), pp. 34–39, 1995.
- [14] T. Tanaka, K. Sakamoto, S. Kusumoto and T. Kikuno, "Improvement of software process by process visualization and benefit estimation," In Proc. 17th International Conference on Software Engineering (ICSE'95), pp. 123–132, 1995.
- [15] E. Yourdon, *Death March : The Complete Software Developer's Guide to Surviving 'Mission Impossible' Projects*, Prentice Hall Computer Books, 1997.