An Empirical Evaluation of Predicting Runaway Software Projects Using Bayesian Classification

Osamu Mizuno[1], Takanari Hamasaki[1], Yasunari Takagi[2], and Tohru Kikuno[1]

[1] Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan,
o-mizuno@ist.osaka-u.ac.jp
[2] OMRON Corporation, 2-2-1 Nishikusatsu, Kusatsu, Shiga 525-0035, Japan
yasunari_takagi@omron.co.jp

Abstract. Since software development projects often fall into runaway situations, detecting signs of runaway status in early stage of development has become important. In this paper, we propose a new scheme for the prediction of runaway projects based on an empirical questionnaire. We first design a questionnaire from five viewpoints within the projects: requirements, estimations, planning, team organization, and project management activities. Each of these viewpoints consists of questions in which experience and knowledge of software risks are included. Secondly, we classify projects into "runaway" and "success" using resultant metrics data. We then analyze the relationship between responses to the questionnaire and the runaway status of projects by the Bayesian classification. The experimental result using actual project data shows that 33 out of 40 projects were predicted correctly. As a result, we confirm that the prediction of runaway projects is successful.

1 Introduction

Recently, software development projects are required to produce highly reliable systems within a short period and at low cost. In other words, software development projects have been put in a very risky situation. Thus, detecting signs of problems at an early stage of the software project is important. If the detection of a problem is delayed, it becomes more difficult to fix a problem since the effort of coping with a problem increases exponentially [1].

Much research has been carried out about the detection of problem signs of a software development project [2,3]. Concerns for risk management are increasing for early detection of such problem signs in software projects. The Software Risk Evaluation method (SRE) is a risk-management technique for a software development project [4]. In the SRE, the project's risks are identified using the taxonomy table of software risks. The risk taxonomy table is very useful for systematically identifying risks of a project. However, since many risk attributes exists in the identification of risks using the taxonomy table, the extraction of a risk takes time. Therefore, SRE recommends to carrying out the tailoring of the taxonomy table for each project.

Risks of a software development project are influenced by environments such as the domain, the business style, the culture of the organization, and by human characteristics. In the projects with similar environments, an approach to prevent recurrence of problems by analyzing past problems is usually taken. Such an approach is easily understood by the project members, since it is based on the actually occurring problem. However, if a problem factor is not arranged appropriately, the number of factors will increase, and handling will become difficult. Efficiency is most important in a project; therefore, software must be developed in a short period of time and at the lowest possible cost. Such efficiency also holds true for project management activity.

This study shows an empirical approach to predict "runaway" projects in an organization. The proposed approach has the following features:

Use of a questionnaire: We utilize a simple questionnaire to identify the characteristics of projects.

Empirical evaluation: We validate our constructed model with actual project data empirically.

Firstly, we investigate the problems on various project using the questionnaire. In developing our questionnaire, we made use of lessons learned from past experience of our projects, and related works in the literature [5–8]. From the results of the questionnaire and past project performance, we identified "runaway" projects and their problems.

Secondly, we analyze the relationship between the responses of the questionnaire and the results of projects using the naive Bayesian classification approach. The naive Bayesian classification is a well-known approach in the data mining, and it is widely used to classify data into several discrete categories. We therefore apply the naive Bayesian classifier to the collected data of the questionnaire in order to classify them into "runaway" and "success"projects.

Finally, we evaluate this approach by the 10-fold cross validation technique. The results of this evaluation sufficiently confirmed the validity of our approach.

2 Background and Objective

The main products of Social Systems Business Company (SSBC) in OMRON consist of embedded software in ticket vending machines, automated teller machines, and in point of sales systems. Such systems for different customers use the same type of hardware and operating systems, but incorporate specific customer requirements, such as customer-specific user interfaces, printed forms, and operation sequences. The development process for these projects is the overlapping waterfall model [9].

In order to promote a process improvement initiative, the Software Engineering Process Group (SEPG) was established in 1992. Various Software Process Improvement (SPI) activities were undertaken by the SEPG, such as;

- Collecting and analyzing a project's actual result data such as quality, cost, and duration
- Monitoring of project status and troubleshooting support
- Developing process standards and procedures in the organization
- Delivering training of the organizational standard process
- Facilitating software process improvement initiatives throughout the organization

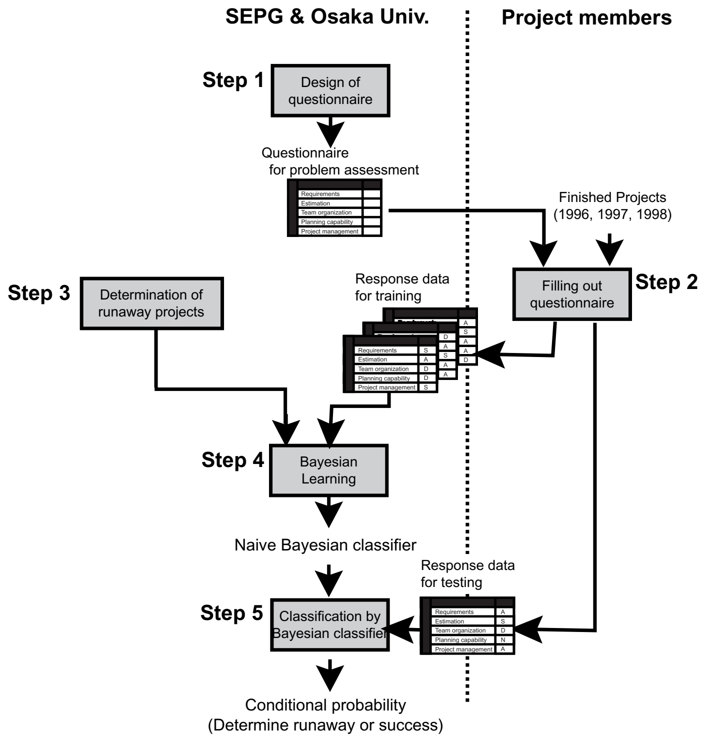

Fig. 1. Outline of prediction of runaway projects

For this reason, the number of runaway projects has decreased yearly. However, the runaway projects still occur every year although the number of them is quite small. In order to reduce such runaway projects, the SEPG has tried to analyze their characteristics. If such characteristics are identified, we can predict runaway-prone projects (also called "risky") and we can deal with such projects before they turn to runaway projects. A similar type of project has been dubbed the *death march project* [10].

In this paper, a "runaway project" is defined as follows:

– A project whose cost and duration is out of a certain range, and,
– A project falling into an uncontrollable situation during development.

3 Outline of Our Approach

Figure 1 shows the outline of our approach for predicting runaway projects. First, in Step 1, we designed a questionnaire to be distributed to project managers and leaders in order to collect the assessment data. Fortunately, in SSBC, actual resultant data and lessons learned for every development project have been stored. The questionnaire consists of five viewpoints each of which is further divided into several risk factors. Next, in Step 2, SEPG distributed the questionnaire to project managers and leaders, and asked them to fill out the questionnaire. After they finished filling out the questionnaire, the SEPG collected them. At the same time, in Step 3, SEPG determined the runaway projects from available project data. Here, we assume that the final status of a project becomes either "runaway" or "success". In Step 4, we apply the naive Bayesian learning to the responses to the questionnaire that is prepared for the model learning, and we obtain the probabilistic model to identify the runaway projects. Finally in Step 5, using the obtained model, we predict the runaway project based on the calculated probabilities.

The questionnaire consists of five viewpoints: requirements, estimations, planning capability, team organization, and project management activities. (The details of the design will be described in Section 5.) Each sub-item regarding risk factors in the questionnaire must be filled in according to the Likert scale [11]: "Strongly Agree", "Agree", "Neither agree nor disagree", or "Disagree".

The naive Bayesian classification is the optimal method of supervised learning if the values of the attributes of an example are independent given the class of the example. Although this assumption is almost always violated in practice, recent research has shown that naive Bayesian learning is also effective in practice.

In our previous research [12], we constructed a logistic model with five candidates used for the parameters. In this paper, we change the modeling technique with more generic model, that is the naive Bayesian classifier, since it is more robust approach to apply more empirical data from various industries.

4 Naive Bayesian Classifier

The naive Bayesian classifier is one of the most common approach to classify categorical data into several classes. We introduce several fundamental concepts briefly.

4.1 Bayesian learning

Let $Q_1, Q_2, \cdots, Q_n$ be the parameters with discrete values to predict a discrete class C. For example, Q_i denotes the questionnaire and C denotes the final status of a project: runaway and success. Suppose that the values $q_1, q_2, \cdots, q_n$ are given to these parameters, and the optimal prediction is the class value $C = \mathrm{runaway}$ such that $P(C = \mathrm{runaway}|Q_1 = q_1 \wedge Q_2 = q_2 \wedge \cdots \wedge Q_n = q_n)$ is maximum. By the Bayes' theorem, this probability is expressed as follows:

$$\frac{P(Q_1 = q_1 \wedge Q_2 = q_2 \wedge \cdots \wedge Q_n = q_n | C = \mathrm{runaway})}{P(Q_1 = q_1 \wedge Q_2 = q_2 \wedge \cdots \wedge Q_n = q_n)} \times P(C = \mathrm{runaway}).$$

The probability $P(C = \mathrm{runaway})$ can be easily estimated from training data. Furthermore, $P(Q_1 = q_1 \wedge Q_2 = q_2 \wedge \cdots \wedge Q_n = q_n)$ is irrelevant to the class variable C. Therefore, learning is reduced to the problem of estimating $P(Q_1 = q_1 \wedge Q_2 =$

$q_2 \wedge \cdots \wedge Q_n = q_n | C = \text{runaway})$ from training data. Using Bayes' theorem again, this conditional probability can be written as follows:

$$P(Q_1 = q_1 | Q_2 = q_2 \wedge \cdots \wedge Q_n = q_n, C = \text{runaway}) \\ \times P(Q_2 = q_2 \wedge \cdots \wedge Q_n = q_n | C = \text{runaway})$$

The second factor of above formula is recursively formulated as follows:

$$P(Q_2 = q_2 | Q_3 = q_3 \wedge \cdots \wedge Q_n = q_n, C = \text{runaway}) \\ \times P(Q_3 = q_3 \wedge \cdots \wedge Q_n = q_n | C = \text{runaway})$$

Here, we assume that Q_i's ($1 \leq i \leq n$) are independent each other. In other words, we assume

$$P(Q_1 = q_1 | Q_2 = q_2 \wedge \cdots \wedge Q_n = q_n, C = \text{runaway}) = P(Q_1 = q_1 | C = \text{runaway})$$

and so on. Then, $P(Q_1 = q_1 \wedge Q_2 = q_2 \wedge \cdots \wedge Q_n = q_n | C = \text{runaway})$ equals

$$P(Q_1 = q_1 | C = \text{runaway}) \times P(Q_2 = q_2 | C = \text{runaway}) \\ \times \cdots \times P(Q_n = q_n | C = \text{runaway}).$$

$P(Q_i | C = \text{runaway})$ ($1 \leq i \leq n$) can be estimated from training data. The process of Bayesian learning is thus performed.

4.2 Bayesian classification

Using the result of learning, we can classify and predict the value of C if the values of q_i are given.

$$P(C = \text{runaway} | Q_1 = q_1 \wedge \cdots \wedge Q_n = q_n) \\ = (\prod_{i=1}^{n} P(Q_i = q_i | C = \text{runaway})) \times P(C = \text{runaway})/z$$

where z is a normalizing constant. We can thus calculate the conditional probability for any given values of parameters q_i's using this equation. From the calculated probability, we classify the data into either class, runaway or success. Here, we determine a project is in class "runaway" if $P(C = \text{runaway} | Q_1 = q_1 \wedge \cdots \wedge Q_n = q_n) \geq 0.5$.

5 Design of the Questionnaire

5.1 Five viewpoints

In this study, we have investigated various works [5–8] regarding risk management and the experience of the SSBC. 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. The overview of the questionnaire is shown in Table 1.

Table 1. Problem Assessment Questionnaire

1. **Requirements**	
1.1	Ambiguous requirements.
1.2	Insufficient explanation of the requirements.
1.3	Misunderstanding of the requirements.
1.4	Lack of commitment regarding requirements between the customer and the project members.
1.5	Frequent requirements or specification changes.
2. **Estimations**	
2.1	Insufficient awareness of the importance of estimation.
2.2	Insufficient skills or knowledge of estimation methods.
2.3	Insufficient estimation for the implicit requirements.
2.4	Insufficient estimation for the technical issues.
2.5	Lack of stakeholders commitment for estimation.
3. **Planning**	
3.1	Lack of management review for the project plan.
3.2	Lack of assignment of responsibility.
3.3	Lack of breakdown of the work products.
3.4	Unspecified project review milestones.
3.5	Insufficient planning of project monitoring and controlling.
3.6	Lack of project members' commitment for the project plan.
4. **Team Organization**	
4.1	Lack of skills and experience.
4.2	Insufficient allocation of the resources.
4.3	Low morale.
5. **Project Management Activities**	
5.1	Lack of resource management of project managers throughout a project.
5.2	Inadequate project monitoring and controlling.
5.3	Lack of data needed to keep track of a project objectively.

5.2 Requirements

The *Requirements* viewpoint includes factors which are related to the understanding and commitment of the requirements among the project members. The factors for the requirements viewpoint are distinguished as follows:

(1.1) Ambiguous requirements
This item checks whether or not the requirements are clear and consistent. It is important for the project members to understand what the customer wants in order to achieve clear and consistent results.

(1.2) Insufficient explanation of the requirements
This item checks whether or not the customers have a sufficient explanation of the requirements regarding the system and/or software.

(1.3) Misunderstanding of the requirements
This item checks whether or not the developers have sufficient skills and/or knowledge to understand the requirements. The developers must have not only sufficient

technical skills and/or knowledge for the project, but must also have specific knowledge regarding the customers' domain.

(1.4) Lack of commitment regarding requirements between the customer and the project members
This item checks whether or not a commitment is obtained by both the project members and the customer. In order to confirm a commitment, it is important to have meetings to review requirements with project members and customers.

(1.5) Frequent requirement changes
This item checks whether or not changes in requirements or specifications are appropriately managed and whether or not project members are kept informed.

5.3 Estimations

The *Estimations* viewpoint includes factors related to the estimation itself, the technical methods for carrying out the estimation, and the commitment between project members and customers. The factors for the estimation viewpoint are distinguished as follows:

(2.1) Insufficient awareness of the importance of the estimation
This item checks whether or not the project members are aware of the importance of estimations. If they are not aware of the importance of the estimation, project members may tend to accept unreasonable requirements.

(2.2) Insufficient skills or knowledge of the estimation method
This item checks whether or not the project members have sufficient skills or knowledge for the estimation methods. In order to show the rationale of estimates, estimation methods must be utilized effectively.

(2.3) Insufficient estimation for the implicit requirements
This item checks whether or not the implicit requirements are considered and estimated. "Must-be" functions in the business area of the customer, or functions implemented in the previous system tend to be the implicit requirements.

(2.4) Insufficient estimation for the technical issues
This item checks whether or not the project members have estimated technical issues sufficiently. For example, technical issues include the selection of the programming language and the development environment.

(2.5) Lack of stakeholders' commitment for estimation
This item checks whether or not the commitment between project members and stakeholders is established. Stakeholders include customers, the sales division, and subcontractors. If a commitment is insufficient and the project members yield to political pressure, unrealistic estimations will be produced.

5.4 Planning

The *Planning* viewpoint includes factors related to the planning or scheduling activity and the commitment for the project plan among project members. The factors for the planning viewpoint are distinguished as follows:

(3.1) Lack of management review for the project plan
This item checks whether or not the project manager reviews the project plan. Management review includes reviewing the project plan to check for feasibility, etc.

(3.2) Lack of assignment of responsibility
This item checks whether or not the project has been systematically divided into activities by using the WBS, and whether or not the responsibility for each technical activity has also been clearly specified. The plan of the project should include not only engineering activities but also project management activities.

(3.3) Lack of breakdown of the work products
This item checks whether or not the work products to be produced by development have been correctly specified. The degree of breakdown should be determined for each development project.

(3.4) Unspecified project review milestones
This item checks whether or not sufficient project review milestones are set up. In the project review, project status, such as progress, quality of work products, etc., is reviewed.

(3.5) Insufficient planning of project monitoring and controlling
This item checks whether or not the plan of monitoring and control the project activities, issues, risks, and work products, is specified correctly.

(3.6) Lack of project members' commitment for the project plan
This item checks whether or not the plan has been reviewed by all of the project members. All engineers engaged in the project must recognize the project plan, and understand the concrete goals of the project.

5.5 Team organization

The *Team organization* viewpoint includes factors related to the staffing of the projects, the fundamental skills and experience and morale of project members. The factors for the team organization viewpoint are as follows:

(4.1) Lack of skills and experience
This item checks whether or not project members have sufficient skills and experience to do their tasks.

(4.2) Insufficient allocation of resources
This item checks whether the resources are well allocated or not.

(4.3) Low morale
This item checks whether the morale of the project members is low or not.

5.6 Project management activities

The *Project management activities* viewpoint includes factors related to the project management activities. The factors which distinguish the project management activities viewpoint are as follows:

(5.1) Project manager lack of resource management throughout a project
This item checks whether or not the project members are actually working on the assigned project. Project managers should act as a firewall so that project members can devote themselves to their tasks.

Table 2. Projects used for our experiment

Projects	1: Requirements					2: Estimations					3: Planning						4: Organization			5: Management			Result
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	6	1	2	3	1	2	3	
PR1	D	D	D	D	D	A	S	S	A	D	A	D	D	D	D	D	A	N	D	D	D	D	Success
PR2	D	D	D	D	D	D	D	D	D	D	D	D	D	A	D	D	D	D	D	D	D	D	Success
PR3	D	D	D	D	S	D	D	A	S	D	D	D	D	D	A	D	D	D	D	D	D	D	Success
PR4	S	S	A	A	S	D	D	A	A	D	A	A	D	D	D	N	A	D	D	D	D	D	Success
PR5	D	D	D	D	A	D	D	D	D	D	D	A	D	A	A	D	D	D	D	D	A	D	Success
PR6	D	S	A	D	D	A	A	A	D	A	D	A	D	D	D	D	D	D	D	D	D	A	Success
PR7	D	D	A	S	A	D	D	D	D	D	D	A	D	S	D	D	D	D	D	D	D	D	Success
PR8	D	A	S	S	D	N	D	A	D	D	A	A	D	D	A	A	D	D	N	S	D	D	Success
PR9	D	A	D	A	S	D	D	D	D	D	A	A	D	A	A	D	D	D	D	D	D	A	Success
PR10	D	D	D	D	A	D	A	A	D	D	D	A	D	D	A	D	D	D	D	A	D	D	Success
PR11	D	S	S	A	D	D	D	S	S	D	D	D	D	D	D	D	D	D	A	D	D	D	Success
PR12	D	A	A	A	D	D	A	D	D	D	D	A	D	A	D	D	D	D	D	D	D	A	Success
PR13	D	A	D	A	D	D	D	D	D	D	A	S	S	D	A	A	A	A	D	A	A	N	Success
PR14	D	D	D	D	S	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Success
PR15	D	A	A	A	A	D	A	A	D	D	D	D	D	D	D	D	S	A	D	S	D	D	Success
PR16	D	D	D	D	A	D	A	D	A	S	S	A	D	A	S	A	S	A	D	A	A	A	Success
PR17	D	D	D	D	D	D	A	D	D	D	A	A	A	S	A	A	D	D	D	A	A	D	Success
PR18	D	D	D	D	N	D	D	D	D	D	D	A	A	D	D	D	D	D	D	D	D	D	Success
PR19	D	D	D	D	S	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Success
PR20	D	A	S	A	S	D	D	D	D	D	S	D	D	D	S	D	A	D	D	D	S	S	Success
PR21	D	A	A	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Success
PR22	S	A	S	S	A	A	N	S	A	N	D	A	A	A	D	N	S	N	A	A	A	D	Success
PR23	A	A	D	A	S	D	D	A	S	D	A	D	A	A	S	A	D	D	D	D	A	S	Runaway
PR24	A	A	S	S	S	A	A	S	A	S	S	S	S	A	S	A	S	S	D	A	A	A	Runaway
PR25	S	A	D	D	S	D	D	D	D	D	S	D	D	S	S	D	D	D	D	D	D	D	Runaway
PR26	D	A	S	A	A	S	D	A	A	N	D	A	D	D	A	A	D	A	A	A	D	A	Runaway
PR27	D	A	A	A	A	D	S	A	S	S	D	A	A	D	D	A	A	A	D	D	D	D	Runaway
PR28	A	S	S	A	A	D	D	S	S	A	S	D	S	D	A	S	A	D	A	D	A	A	Runaway
PR29	S	A	S	A	D	S	A	A	A	D	D	A	A	A	S	D	A	D	A	D	S	S	Runaway
PR30	A	A	S	S	A	D	D	A	D	A	A	A	A	A	A	D	S	D	A	D	A	D	Runaway
PR31	D	D	D	D	D	D	D	D	D	D	S	S	S	S	S	S	S	S	D	D	S	S	Runaway
PR32	A	S	S	S	A	A	S	S	S	S	S	S	S	A	S	S	S	S	A	S	S	D	Runaway
PR33	A	D	D	D	D	D	A	D	A	D	D	S	A	D	D	A	S	D	D	A	D	D	Success
PR34	D	D	D	D	D	D	D	D	D	D	D	A	D	D	D	D	D	D	D	D	D	A	Success
PR35	A	A	A	A	S	D	A	D	D	D	D	A	A	D	D	D	D	D	D	D	D	D	Success
PR36	D	D	D	D	A	D	D	D	A	A	A	A	D	D	D	D	D	D	D	D	D	D	Success
PR37	D	D	D	D	N	D	D	D	D	D	D	A	D	D	D	D	D	D	D	D	D	D	Success
PR38	S	D	D	A	A	D	A	A	S	S	D	A	A	D	D	D	A	S	D	S	S	D	Runaway
PR39	A	S	S	S	A	N	N	S	N	N	N	S	S	S	D	A	A	N	S	D	D	D	Runaway
PR40	A	A	A	A	S	A	D	S	D	D	S	S	S	S	S	S	D	D	D	D	A	A	Runaway

(5.2) Inadequate project monitoring and controlling

This item checks whether or not progress monitoring is adequately done, whether or not the progress reporting is actually done, and whether or not corrective action is adequately taken.

(5.3) Lack of data needed to keep objective track of a project

This item checks whether or not the project managers are able to objectively keep track of a project on the basis of the software metrics collected during development. If such data are not available, then it is difficult to recognize the project status correctly to make management decisions appropriately.

6 Application of the Bayesian Classification

In this section, we apply the proposed questionnaire in Section 5 to sample projects, and successively apply the naive Bayesian classifier to the assessment data obtained from the questionnaire.

6.1 Filling out the questionnaire

We chose 40 projects, which were part of the projects performed from 1996 to 1998 by the SSBC. The SEPG distributed the questionnaires designed in Section 5 to the project managers or the project leaders of 40 target projects, and explained the details of the questionnaire and the purpose of the trial. The responses to the questionnaire were collected by the SEPG after one month. Table 2 shows the collected responses. In Table 2, the answers “Strongly agree”, “Agree”, “Neither agree nor disagree”, and “Disagree” are shown as characters “S”, “A”, “N”, and “D”, respectively.

6.2 Determination of runaway projects

Since all of these projects completed their development, the SEPG had already identified the runaway projects according to the decision process mentioned in Section 3. As a result, 13 projects out of 40 were classified as runaway projects. Thus, the column, “Result”, in Table 2, shows the actual result of the classification.

6.3 Prediction by Naive Bayesian Classifier

We then applied the naive Bayesian classifier to the response data of all projects, and calculated the conditional probabilities of being runaway. On applying the naive Bayesian classifier, we assume that each response of question is a nominal value.

In order to show the effectiveness of the proposed approach, we perform 10-fold cross validation on the collected data. The evaluation through k-fold cross validation method is one of the most common in machine learning community. The data set is here split into k equally sized subsets, and then in i-th iteration ($i = 1 \cdots k$) i-th subset is used for testing the classifier that has been learned from all other remaining subsets. Notice that each instance of data is classified exactly once. The number of subsets k is usually set to 10.

Let us show an example of 10-fold cross validation. At first, we randomly split 40 projects in Table 2 into 10 subsets. Suppose that a subset contains projects PR3, PR12, PR37, and PR39. Next, we select this subset for testing, and thus rest 9 subsets are used for Bayesian learning. Four projects are then classified by the learned Bayesian model. For each subset, the same procedure is performed. Finally, we obtain 40 testing results to evaluate the overall accuracy of the prediction.

Table 3. Results of 10-fold cross validation

	Predicted	
Actual	Success	Runaway
Success	22	5
Runaway	2	11

Table 3 shows the result of 10-fold cross validation. The rows show the number of projects that are actually runaway or success. The columns show the number of projects that are predicted as runaway or success.

As shown in Table 3, 33 (that is, 22+11) out of 40 projects can be predicted correctly. The predicting accuracy is thus 82.5%. According to Fisher's exact test, we can say that actual and predicted results are not independent with statistical significance $\alpha = 0.01$. We can conclude that it is rather high accuracy.

This result indicates that the prediction works well irrelevant to the selection of projects. For this result, we can say that our proposed approach can be applicable to the prediction of runaway projects.

7 Conclusion

In this research, we tried to construct prediction model of the runaway projects using the naive Bayesian classification method. We showed that a naive Bayesian classification model can be applicable to the response data of questionnaire and the prediction can be done with high accuracy. Since the data used in this study are the information regarding the final status of projects and the responses of the questionnaire about the software risks, this data can be collected easily in actual development fields.

As future work, we will apply the proposed approach to other data set obtained from several companies. Furthermore, in order to make the prediction more accurate, we have to seek the other classification methods suitable for the software engineering.

Acknowledgments

The authors would like to thank Dr. Keishi Sakamoto for his support to our entire process of improvement activities. The authors also would like to express great thanks to the project members of the 40 projects who willingly cooperated with our requests.

References

1. Boehm, B.W.: Industrial software metrics top 10 list. IEEE Software **4** (1987) 84–85
2. Jiang, J., Klein, G.: Software development risks to project effectiveness. Journal of Systems and Software **52** (2000) 3–10
3. Wohlin, C., Andrews, A.A.: Prioritizing and assessing software project success factors and project characteristics using subjective data. Empirical Software Engineering **8** (2003) 285–303
4. Williams, R.C., Pandelios, G.J., Behrens, S.G.: Software risk evaluation (SRE) method description (version 2.0). Technical Report CMU/SEI-99-TR-029, Software Engineering Institute (1999)
5. Conrow, E.H., Shishido, P.S.: Implementing risk management on software intensive projects. IEEE Software **14** (1997) 83–89
6. Fairley, R., Rook, P.: Risk management for software development. In: Software Engineering. IEEE CS Press (1997) 387–400
7. Karolak, D.W.: Software Engineering Risk Management. IEEE CS Press, CA (1996)
8. Sisti, F.J., Joseph, S.: Software risk evaluation method version 1.0. Technical Report CMU/SEI-94-TR-19, Software Engineering Institute (1994)
9. Humphrey, W.S.: A Discipline for Software Engineering. Addison-Wesley, MA (1995)

10. Yourdon, E.: Death March : The Complete Software Developer's Guide to Surviving 'Mission Impossible' Projects. Prentice-Hall Computer Books (1997)
11. Fenton, N.E., Pfleeger, S.L.: Software Metrics : A Rigorous & Practical Approach. PWS Publishing (1997)
12. Mizuno, O., Kikuno, T., Takagi, Y., Sakamoto, K.: Characterization of risky projects based on project managers' evaluation. In: Proc. of 22nd International Conference on Software Engineering. (2000) 387–395