An integrative framework for intelligent software project risk planning

Yong Hu a,⁎, Jianfeng Du a, Xiangzhou Zhang b, Xiaoling Hao c, E.W.T. Ngai d, Ming Fan e, Mei Liu f

a Institute of Business Intelligence and Knowledge Discovery, Guangdong University of Foreign Studies, Sun Yat-sen University, Guangzhou 510006, PR China
b School of Business, Sun Yat-sen University, Guangzhou 510006, PR China
c Shanghai University of Finance and Economics, Shanghai 200433, PR China
d Department of Management and Marketing, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China
e Foster School of Business, University of Washington, Seattle, WA 98195, USA
f Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102, USA

Available online 30 December 2012

Keywords:
Decision support systems
Software project risk analysis and planning
Actionable knowledge discovery
Social media platform project

Abstract
Software projects have inherent uncertainties and risks. Social software projects suffer even more requirement changes and require more attention to risk management. Risk analysis and planning are complex, making it difficult to manage risks effectively through subjective judgment. At present, ample empirical research on intelligent decision-support models for risk analysis in software projects exists. However, to the best of our knowledge, empirical models for software project risk planning, or those related to integrative software risk analysis and planning are not available. Thus, the current study proposes an integrative framework for intelligent software project risk planning (IF-ISPRP) to help in minimizing the impacts of project risks and achieving a better foreseeable project outcome. IF-ISPRP includes two core components, namely, risk analysis module and risk planning module. The risk analysis module is to predict whether a project will be successful or not. The risk planning module is to produce a cost-minimal action set for risk control based on the risk analysis module. For integrative risk analysis and planning, we propose a novel many-to-many actionable knowledge discovery (MMAKD) method for complex risk planning. We also apply the framework on a social media platform project, Guangzhou Wireless City, and demonstrate how the model can generate a cost-minimal action set to mitigate the project risk. The risk-control actions found may help develop strategies on mitigating the risks of other social software projects. We hope that the proposed framework will provide an intelligent decision-support tool for project stakeholders to effectively control project risks by integrating risk analysis and planning.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Software projects have inherent uncertainties and risks [32]. Moreover, software development has a high failure rate. The Standish Group [46], in their report “CHAOS Summary 2009,” showed that the success rate of global software projects is only approximately 32%. Software development is subject to a variety of risks, which are generally defined as a set of factors or conditions that seriously threaten the successful completion of a software project [48]. The necessity of identifying and tackling these risks has spurred significant interest among academic researchers and practitioners [6]. A research by the Microsoft corporation indicated that putting 5% work into effective risk management could produce a 50%–75% opportunity to complete project before the deadline [36]. It helps to prevent catastrophes such as redo and overkill of a software project.

“Social software” has been used to denote Internet software that enables people to communicate and collaborate, such as e-mail, IM, group forums, blogs, and SNS [11]. The designers and developers of social software are faced by a major challenge — how to facilitate social interaction and engagement among the users [22]. Although social software projects share common risks with general software projects, they also have their own distinct and unique risk source because people use social software for a variety of purposes [22]. Moreover, as a common characteristic of social software, the continuous and fast requirement changes, in terms of user interface design and interaction functions, make it more imperative to conduct risk management.

Risk analysis and planning are two of the most important processes in software project risk management (SPRM). Risk analysis involves exploring the relationships between risk factors and project outcome [1,43], which is critical for effective risk control. Substantial research on risk analysis models based on intelligent technologies is available such as [27,30,31,37,49]. Note that this paper does not aim to discuss the topics of software reliability, safety, cost/effort, etc., such as those mentioned in [7]. Risk planning involves developing an effective plan to address the most critical risks and optimizing the allocation of project resources [1,43]. While risk planning is generally acknowledged as critical to risk
management, few studies have been put forward and there is no widely accepted tool or model to support it [6].

At the U.S. National Aeronautics and Space Administration (NASA), risk planning requires not only “developing effective plans” but also planning “only as much as needed” [43]. Boehm [9] believes that risk planning should not only “plan for each risk item” but also be able to “integrate the risk-management plans for each risk item with each other and with the overall project plan.” Therefore, in order to optimize resource allocation while maximizing the project outcome, it is imperative to consider the combined impact of the risk factors on the project outcome as well as the total execution cost of the selected risk-control actions. Due to the tremendous computational effort, intelligent decision-support models/tools are preferred over subjective expert judgment in achieving effective and economical decision making during risk analysis and planning. Studies that adopt integer-programming technology for risk planning, such as those of Ben-David and Raz [6] and Pan and Chen [40], aim to eliminate all kinds of risks. However, in a resource-restricted situation, such objective is not practical because of the high cost of controlling all the risks. Moreover, controlling all risks is often not necessary as only a few risks can directly affect the project outcome. Furthermore, the risk of continuous requirement changes cannot be avoided in a social software project. These studies cannot identify the key risks because they have not taken advantage of the relationships between risk factors and project outcome, which are characterized by an intelligent risk analysis model.

To overcome the above limitations, the present study proposes an empirical and intelligent framework that can provide an integrative risk analysis and planning in SPRM. We first introduce the components and application steps of the framework. Given a specific project, the project outcome is predicted and then, based on the prediction mechanism and planning context, a cost-minimal risk-control plan is determined. Random forest [12] is adopted to establish a risk analysis model while and planning context, a cost-minimal risk-control plan is determined. The project outcome is predicted and then, based on the prediction mechanism and planning context, a cost-minimal risk-control plan is determined. Therefore, in order to optimize resource allocation while maximizing the project outcome, it is imperative to consider the combined impact of the risk factors on the project outcome as well as the total execution cost of the selected risk-control actions. Due to the tremendous computational effort, intelligent decision-support models/tools are preferred over subjective expert judgment in achieving effective and economical decision making during risk analysis and planning. Studies that adopt integer-programming technology for risk planning, such as those of Ben-David and Raz [6] and Pan and Chen [40], aim to eliminate all kinds of risks. However, in a resource-restricted situation, such objective is not practical because of the high cost of controlling all the risks. Moreover, controlling all risks is often not necessary as only a few risks can directly affect the project outcome. Furthermore, the risk of continuous requirement changes cannot be avoided in a social software project. These studies cannot identify the key risks because they have not taken advantage of the relationships between risk factors and project outcome, which are characterized by an intelligent risk analysis model.

To overcome the above limitations, the present study proposes an empirical and intelligent framework that can provide an integrative risk analysis and planning in SPRM. We first introduce the components and application steps of the framework. Given a specific project, the project outcome is predicted and then, based on the prediction mechanism and planning context, a cost-minimal risk-control plan is determined. Random forest [12] is adopted to establish a risk analysis model while a many-to-many actionable knowledge discovery (MMKAID) method is proposed to accomplish risk planning. Finally, we implement the proposed framework and present a real case study of social media platform project to demonstrate the application of the proposed framework.

The study has two specific contributions. First, the proposed framework for integrative risk analysis and planning is the first in the field of SPRM. Second, the proposed risk-planning method considers many-to-many relationships between risk factors and risk-control actions, which adequately meet the requirements of risk analysis and planning in practice. A many-to-many relationship implies that a risk factor can be influenced by several risk-control actions, and vice versa a risk-control action can influence several risk factors. In the case study, the social media platform project (Guangzhou Wireless City) demonstrates how the model generates a cost-minimal action set to mitigate the whole project risk.

2. Literature review

2.1. Software project risk management

Risk management was introduced into software project management by Boehm [8] and Charette [16]. SPRM is a series of rules or practices, which can identify, analyze, and monitor the risk factors as well as increase the success rate of the project [3]. SPRM could positively influence budget, schedule, scope of the project, etc. [28]. In general, SPRM comprises two stages [8,16,25]:

1) First stage is risk assessment which involves risk identification, risk analysis, and risk prioritization. Risk identification requires systematic identification and classification of the risk factors. Risk analysis assesses the state of each identified risk factor, and analyzes the relationships among risk factors, and between risk factors and project outcome. Risk prioritization decides the priority sequence in controlling each risk factor.

2) Second stage is risk control which involves risk planning and risk monitoring. Risk planning involves not only planning for each risk factor, but also coordinating the individual plans with each other. Continuous monitoring of the states of risk factors, examination of the effectiveness of the risk-control plan, and the prompt discovery of impending risks are required during and after the implementation of the plan.

2.2. Software project risk analysis and planning

Recently, a significant number of related studies on risk analysis and planning in the field of SPRM have become available. This paper focuses on historical-data-based risk analysis, specifically on predicting the probability of success for software development projects; thus we exclude studies that are based on subjective analysis or expert judgment, which are methods often used in project risk management [19]. We also exclude studies that analyze software reliability, cost, safety, etc. The representative literature is summarized in Table 1. To the best of our knowledge, there is no specific study on the topic of intelligent risk management for social software project.

Risk analysis is more widely studied and the modeling methods mainly include the use of statistical analysis and data mining.

1) Research based on statistical methods. For example, Wallace et al. [48] used structural equation modeling (SEM) to develop an exploratory model for testing and estimating the relations between software project risks and project performance. Drew Procaccino et al. [17] assessed several early risk factors and their effects on software project success using regression analysis. Their research found that the presence of a committed sponsor and the level of confidence that the customers and users have in the project manager and the development team are the most important factors for project success. Jiang and Klein [27] presented a model using principal component analysis (PCA) based on a survey of 86 project managers, exploring the relationships between IS success measures and risk factors. In conclusion, these studies aim to discover the universal knowledge about risks (i.e. the correlation or causality between risk factors and project outcome) rather than predict the overall risk level of an ongoing project.

2) Research based on data mining methods. For example, Aguilar-Ruiz et al. [2] proposed an approach using a software project simulator and evolutionary computation. Their model predicts whether a project can be kept within the cost, quality and duration targets. Zhang et al. [53] built an early warning system to predict project escalation. They found that neural network is preferable to logistic regression model in prediction, as it can capture the non-linear relationships between risk factors and project outcome. However, the network is a “black box”, and thus cannot provide explicit decision-making knowledge to managers. Fan and Yu [21] presented a risk analysis model using Bayesian networks. Lauria and Duchessi [31] demonstrated a methodology for building a Bayesian network for information technology (IT) implementation from survey data and used it to predict the attainment of IT benefits. Moreno García et al. [37] used association rules technology to evaluate the impact of certain project management policies on software project quality, duration, and development effort. In conclusion, the models based on data mining methods can analyze and predict dynamic risk level of ongoing projects.

On the other hand, there are relative few studies on risk planning. They can be divided into two types.

1) Studies that adopt statistical methods to verify the effectiveness of risk-control actions. For example, Benaroch et al. [5] used logistical regression analysis to verify the validity of the risk-option mappings of an IT investment option-based risk management (OBRIoM) framework. Li et al. [32] adopted Pearson’s correlation to analyze the
relationship between risk-reduction activities and risk factors. Jiang et al. [29] found that user partnering is significantly relevant to higher user support, less residual risk, and better project performance. Na et al. [38] proved the presence of a strong negative relationship between standardization and residual performance risk. Li et al. [32] validated and compared the effects of five risk-reduction activities on risk factors based on 133 software projects. Wang et al. [50] collected data from 212 project managers from the Project Management Institute (PMI) and verified that the level of control activities during the system development process has a significant effect on software flexibility. In conclusion, statistical methods are often applied to study the validity of risk-control actions on risk factors.

2) Studies that aim to produce the optimal risk-control action set using integer programming technology. For example, Ben-David and Raz [6] built a risk-response model, which can represent the overlapping effects of multiple risk reduction actions and the impacts of secondary risk events. This risk-response model can also evaluate the total risk exposure under various combinations of risk reduction actions. The model applies integer-programming technology to generate the most cost-effective combination of risk reduction actions. Pan and Chen [40] presented an economic optimization model for selecting risk reduction actions in the risk response planning phase of CMMI-based SPRM. Their model considers the logical constraints (exclusion and implication) that limit the combination of reduction actions, and can be solved using integer-programming technology. Based on a real option analysis framework, Wu et al. [51] applied multistage stochastic integer programming to address complex decision-making problems in ERP project investment.

3. Integrative framework for intelligent software project risk planning

To achieve an integrative risk management that can provide decision support from risk analysis to planning, we propose an integrative framework for intelligent software project risk planning (IF-ISPRP). As shown in Fig. 1, our framework is composed of three key components: (1) Project Risk Database, (2) Risk Analysis Module, and (3) Risk Planning Module.

3.1. Key components of the proposed framework

**Project Risk Database** is a collection of risk factors and final outcomes of real software projects. It provides project samples for the risk analysis module. The sample size and quality significantly influence the validity and reliability of the risk analysis module and subsequently, of the risk planning module.

**Risk Analysis Module** takes a risk analysis model as its core to analyze and predict the project risks. The model takes the states of risk factors as input and returns the predicted project outcomes (i.e., success or failure) as output.

**Risk Planning Module** takes the risk-control actions list, the many-to-many relationship between actions and risk factors, and the execution cost of actions as input. It outputs a cost-minimal risk-control action set that can help in achieving that the prediction of the project outcome is success. The module proceeds through a generate-and-test-based process as follows:

1) A new candidate action set is generated according to the list of risk-control actions, the execution costs of the actions, and the many-to-many relationship between risk-control actions and risk factors.

2) The effect of the candidate action set on the states of risk factors is calculated. Given the new states, the risk analysis module predicts the project outcome. If success is predicted, the total execution cost of the candidate action set is compared with that of the current optimal set. The less costly one is selected as the new optimal set.

3) If having traveled the entire search space, the module outputs the current optimal set, which is a cost-minimal action set. Otherwise, it proceeds to the first step.

3.2. Steps of applying the proposed framework

As shown in Fig. 2, IF-ISPRP application comprises five steps. The first three steps focus on building a reusable tool/model for integrative risk analysis and planning whereas the last two steps are to apply the tool to an ongoing software project.

**Step 1 Data collection**

First, the risk factors that will be used in the risk analysis module are identified. Second, project samples are collected according to the list of risk factors to construct the project risk database. Collecting high-quality software project samples is a long-term and expensive process.

**Step 2 Risk analysis module construction**

First, a proper modeling method for risk analysis is chosen by considering the characteristics of the collected project samples and the interpretability of the objective model. Second, a risk analysis model is constructed based on the training data provided by the project risk database.

**Step 3 Risk planning module construction**

First, the many-to-many relationship model between risk-control actions and risk factors is established. The model is to restrict the scope of the risk planning problem covered in the present study. Second, the proposed MMAKD...
The method is implemented to solve the risk planning problem (details of both the model and the method are presented in Section 4).

**Step 4 Risk analysis**
First, the current states of the risk factors of the project are assessed. Second, the risk analysis module is used to predict the project outcome. If the predicted outcome is a failure, the subsequent risk planning is pursued. The procedures of risk analysis should be periodically performed.

**Step 5 Risk planning**
First, the parameters of risk planning, including a list of candidate risk-control actions, the many-to-many relationship between

---

![Integrative framework for software project risk intelligent planning.](image)

**Fig. 1.** Integrative framework for software project risk intelligent planning.

![Application steps of IF-ISPRP.](image)

**Fig. 2.** Application steps of IF-ISPRP.
actions and risk factors, and the execution costs of actions are assessed. Second, taking the above parameters as input, the risk planning module outputs the cost-minimal action set. Third, the project stakeholders evaluate the generated action set and modify it according to real-life situations. Consequently, the risk resolution process is initiated to implement the produced plan.

4. Many-to-many actionable knowledge discovery

4.1. Problem setting: formulation of the risk planning problem

Some notations are first defined as follows. \( \{F_1, \ldots, F_n\} \) — a set of features (feature refers to risk factors in the present study); \( \{A_1, \ldots, A_m\} \) — a set of actions; \( \langle v_1, \ldots, v_n \rangle \) — a state of the given instance, where \( v_i \) is the value of feature \( F_i \); \( \langle v_1, \ldots, v_n \rangle \) — a set of all possible values of feature \( F_i \); \( \langle c_1, \ldots, c_m \rangle \) — a vector of nonnegative execution costs, where \( c_i \) denotes the execution cost of action \( A_i \), particularly \( c_i = 0 \) denoting no execution. 

\[ f_i(c_i, v_i) \] — a function that returns the new value of feature \( F_i \) obtained from the value \( v_i \) by solely applying action \( A_i \) with an execution cost \( c_i \); 

\[ f_i(C, v_i) \] — a function that returns a new value of feature \( F_i \) obtained by applying actions at costs that represented by \( C = \langle c_1, \ldots, c_m \rangle \): 

\[ \sum_{i=1}^{m} c_i = \text{minimum} \] among all vectors, \( C = \langle c_1, \ldots, c_m \rangle \); 

\[ C = \langle c_1, \ldots, c_m \rangle \] — and such that the given instance with a new state represented by \( \langle f_1(C, v_i), \ldots, f_n(C, v_i) \rangle \) is classified to the preferred output class \( 0_p \). 

4.2. Problem setting: the many-to-many relationship model

Most existing studies on actionable knowledge discovery [33,52,53] only deal with the one-to-one relationship between actions and factors. In the present study, we consider a more general and reasonable case, i.e., the many-to-many relationship between actions and risk factors. In the present study, we consider a more general and reasonable case, i.e., the many-to-many relationship between actions and risk factors. The many-to-many relationship acts as follows:

**Assumption 1.** Actions are applied synchronously; the dependencies and execution orders among actions are not considered.

**Assumption 2.** The compound effect of a set of actions is equal to the maximum effect of an individual action. Thus, we have

\[ f_i(c_1, \ldots, c_m, v_i) = \max_{1 \leq 1 \leq m} f_{ij}(c_j, v_i) \quad \text{for all} \ 1 \leq i \leq n. \] (1) 

For example, let \( A_1 \) and \( A_2 \) be two actions that will change the value of factor \( F \) from high (risk level) to mid and to low respectively. When both \( A_1 \) and \( A_2 \) are applied, the value of \( F \) would be changed from high to low (noting that, we use larger numeric value for lower risk level).

**Assumption 3.** The effect of an action is defined using a monotonically increasing step function on the execution cost of the action.

Hence, for any action \( A_i \) and any value \( v_j \) of \( F \) where \( 1 \leq k \leq n \), there exists an increasing sequence of positive thresholds \( \delta_{ij,v_k,v_{k+1}}, \ldots, \delta_{ij,v_{n},v_{n+1}} \) such that

\[ f_{ij}(c_j, v_i) = \begin{cases} v_k & \text{if } 0 \leq c_j < \delta_{ij,v_k,v_{k+1}} \\ v_{k+1} & \text{if } \delta_{ij,v_k,v_{k+1}} \leq c_j < \delta_{ij,v_{k+1},v_{k+2}} \\ \vdots & \text{for all } k < n \\ v_n & \text{if } c_j \geq \delta_{ij,v_{n-1},v_{n}} \end{cases} \] (2) 

For example, suppose the cost threshold for changing a risk factor \( F \) from high to mid by action \( A_1 \) is $2$, then \( A_1 \) cannot change the value of \( F \) from high to low with a cost less than $2$ and, with a cost more than $2$, \( A_1 \) will do no worse than change the value of \( F \) from high to mid.

4.3. Problem solving: a MMAKD method based on random forest

The above risk planning problem is a search problem and can be solved through a traditional method which is based on the exhaustive generate-and-test framework. In this method, all possible vectors of execution costs are generated and only the cost-minimal one that can make the given instance to be classified to the preferred output class is chosen. However, the method is relatively inefficient when the search space is large and the classifier is complex, because the given classifier is applied as a black box and the classification process cannot be done in a heuristic在市场上-prune the instance search space. To improve the efficiency, we consider encoding the above problem as a propositional satisfiability (SAT) problem so that modern powerful SAT techniques can be used to direct the search and prune the search space. Therefore, we develop a novel method, called the MMAKD method, to extract a cost-minimal action set from random forest [12]. The key idea is to reduce the original problem into an extended SAT problem, called linear PBO problem [35], defined as follows:

\[
\begin{align*}
\text{minimize} & \quad \sum_{j=1}^{n} c_j x_j \\
\text{subject to} & \quad \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad a_{ij}, b_i \in \mathbb{Z}, \ i \in \{1, \ldots, m\},
\end{align*}
\]

where \( c_j \) is a nonnegative integer associated with the variable \( x_j \), and \( a_{ij}, \ldots, a_{im} \) are integer coefficients of \( x_j \) in the set of \( m \) pseudo-Boolean constraints. A 0–1 truth assignment \( \Phi \) on \( \{x_1, \ldots, x_m\} \) is referred to as a solution to the above linear PBO problem if \( \sum_{j=1}^{n} a_{ij} \Phi(x_j) \) is minimized among all 0–1 truth assignments \( \Phi \) such that \( \sum_{j=1}^{n} a_{ij} \Phi(x_j) \geq b_i \) holds for all \( 1 \leq i \leq m \). A linear PBO problem can be solved by efficient pseudo-Boolean solvers, such as MiniSAT+ [20], GALENA [15], and PUEBLO [45].

The proposed MMAKD method uses a random forest as the classifier, i.e., the project outcome predictor. Random forest [12] is an ensemble classifier, which takes decision trees as base classifiers. A decision tree is an analytical decision support tool that uses a tree-like graph to solve classification problems. Random forest uses a majority voting mechanism for aggregating individual outcomes of decision trees. A random forest can be easily encoded as a set of (classification) rules. The reasons why rule representation is adopted would spontaneously show up later. The method is outlined as follows.

Initially, a set of variables is introduced in the target linear PBO problem. Variables of the form \( x_{ij} \) are introduced to denote whether action \( A_i \) is applied with an execution cost \( c \). The variable \( x_{ij} \) is introduced only when applying action \( A_i \) with an execution cost \( c \) that is sufficient to change the original value \( v_{ij} \) of \( F_i \) to a larger value. For every action \( A_i \), a
variable \( x_{0,j} \) is also introduced to denote whether action \( A_j \) is not applied. Variables of the form \( y_{i,v_j} \) are introduced to denote whether the value of \( F_i \) is changed to \( v_{j,k} \) by applying all actions. Variables of the form \( z_{i,j,v_{j,k}} \) are introduced to denote whether the value of \( F_i \) is changed to \( v_{j,k} \) by applying action \( A_j \). The variable \( w_{i,v_{j,k}} \) is introduced only when \( v_{j,k} \geq v_{i,k} \), because we assume that the value of any feature cannot be decreased by applying actions. Moreover, variables of the form \( w_{i,v_{j,k}} \) are introduced to denote whether the \( i \)th decision tree assembled in the random forest can classify the given instance with the new state to the class \( c_{o_i} \). The variable \( w_{i,v_{j,k}} \) is introduced for every decision tree and every possible output class of the given random forest.

Subsequently, all pseudo-Boolean constraints in the target linear PBO problem are generated using the sub-procedures shown in Fig. 3. For convenience, standard rule form is adopted to represent the constraints, using symbols including contradiction \( \bot \), conjunction \( \wedge \), implication \( \rightarrow \) and equivalence \( \leftrightarrow \). Let \( \Pi \) denote the set of these pseudo-Boolean constraints that can be represented as rules.

The encoding of the computation of \( f_1(C, v_{1,k}, \ldots, v_{n,k}) \) for a vector of execution costs \( C = < c_1, \ldots, c_m> \) yields two subsets of \( \Pi \):

1) The first subset consists of rules that encode action constraints and is appended to \( \Pi \) in the sub-procedure "EncodeActionConstraints." The rules added in line 2 specify that every action is applied with some execution cost or is not applied. The rules added in line 3 specify that an action cannot be applied with two different execution costs.

2) The second subset consists of rules that encode action–feature interaction constraints and is appended to \( \Pi \) in the sub-procedure "EncodeFeatureConstraints." The rules added in line 2 specify that for every feature, the original value should be changed to some larger value or remain unchanged after all actions are applied. The rules added in line 3 stipulate that the original value cannot be changed to two different values. The rules added in line 4 specify that the new value obtained by applying all actions is equal to the maximum value obtained by applying individual actions. These rules encode the assumption given in Formula (1). The rules added in line 5 state that for every feature, a new value can be obtained by applying a single action only when the corresponding threshold is finite, and the new value can only be obtained when the execution cost is not less than the corresponding threshold but is less than the next threshold. These rules encode the assumption given in Formula (2).

Suppose the given random forest is assembled with \( t \) decision trees \( T_1, \ldots, T_t \), then the encoding of the condition that \( f_1(C, v_{1,k}, \ldots, v_{n,k}) \) is classified to the preferred output class \( c_{o_p} \) yields another two subsets of \( \Pi \):

1) The first subset consists of rules that encode the classification process of every decision tree and is appended to \( \Pi \) in the sub-procedure "EncodeDecisionTrees." It is well-known that a decision tree can be translated into a set of rules, where each rule corresponds to a path through the tree.
from the root to a leaf of the decision tree. Thus, line 2 adds to \( I \) all the rules translated from the paths that match a possible new state of the given instance, while line 3 adds to \( I \) the rules that ensure the given instance with the new state to be classified to a unique class by an individual decision tree.

2) The second subset consists of pseudo-Boolean constraints that encode the output class of the given random forest and is appended to \( I \) in the sub-procedure “EncodeVotingResult.” The constraints added in line 1 state that the mode of the class outputs of all decision trees assembled in the random forest is unique and is exactly the preferred class \( o_p \).

Finally, rules in \( I \) are translated to a set of pseudo-Boolean constraint. Each rule can be translated to a set of logically equivalent SAT clauses [20] using the mapping function \( \pi \). For example, 
\[
\pi(C_1 \land C_2) = [C_1, C_2].
\]
\[
\pi(x_1 \rightarrow x_2) = [\neg x_1 \lor x_2].
\]
\[
\pi(x_1 \rightarrow \bot) = [\neg x_1].
\]
where \( C_1 \) and \( C_2 \) are standard SAT clauses, and \( x_1 \) and \( x_2 \) are variables. Moreover, SAT clauses of the form, 
\[
\neg x_1 \lor \ldots \lor \neg x_p \lor x_{p+1} \lor \ldots \lor x_q \quad (\text{where } 0 \leq p \leq q)
\]
are logically equal to a pseudo-Boolean constraint 
\[
-1 \cdot x_1 + \ldots + -1 \cdot x_p + 1 \cdot x_{p+1} + \ldots + 1 \cdot x_q \geq 1 - p \quad (\text{where } 0 \leq p \leq q).
\]
Therefore, the target linear PBO problem is formed by combining the set of pseudo-Boolean constraints with the object function \( \sum x_{j,c} \leq x_{j,c} \) to be minimized. Each solution of the target linear PBO problem is a 0-1 truth assignment \( \Phi \) on all variables appearing in \( I \). It corresponds to a cost-minimal action set. That is, for all \( 1 \leq j \leq m \), let \( c_j \) be the unique execution cost of action \( A_j \) such that \( \Phi(x_{j,c}) = 1 \), then \( C = < c_1, \ldots, c_m > \) is a vector of execution costs that the original problem aims to compute.

5. Implementation of IF-ISPRP

5.1. Data collection

Researchers have proposed a variety of classification frameworks on software project risk factors. For example, Barki et al. [4] divided 35 software risks into five dimensions, including technological newness, application size, expertise, application complexity, and organizational environment. Schmidt et al. [44] launched a cross-cultural research in Hong Kong, Finland, and the U.S. They established a framework that covers 14 dimensions and 33 risks. Wallace et al. [48] classified 27 software risk factors into six dimensions, as shown in Table 2. The current paper adopted the framework proposed by Wallace et al. because: 1) it not only systematically summarizes the previous studies but it also offers extensions; 2) it is relatively new, and it has been frequently cited for further investigation, e.g. in Han and Huang [24] and Huang and Han [26].

With the support of the Guangdong Software Industry Association, a total of 460 questionnaires were sent out and the respondents were requested to fill the questionnaires based on their recent software project. A total of 317 respondents (response rate of 69%) returned the questionnaires, out of which 269 were valid responses. Questionnaires missing any of the risk factor values were considered incomplete or invalid and thus excluded. Results showed that 27.1% of the projects were successful, while others were unsatisfactory or canceled. The surveys covered a wide variety of sectors such as government (15.2%), information industry (26%), manufacturing (11.9%), and commerce (15.2%). More than 80% of the respondents had related work experience of more than three years. The respondents were project managers (27.5%), project technical leaders (19.0%), development team members (34.9%), etc.

5.2. Module construction and evaluation

A risk analysis model requires method with good interpretability and high predictive accuracy. Numerous algorithms are suitable, such as the decision tree, the Bayesian network, etc. The present study adopts random forest algorithm which was usually shown to be more accurate than decision tree and is consistent with the proposed MMAKD method.

In this study, successful projects are regarded as positive samples, whereas failed ones are regarded as negative samples. We denote as \( TP \) the number of true positive samples (i.e., positive samples that are classified as positive), \( TN \) as the number of true negative samples, \( FP \) as the number of false positive and \( FN \) as the number of false negative samples. To evaluate the model’s performance, the following four measures are calculated.

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}
\]
\[
\text{Precision} = \frac{TP}{TP + FP}
\]
\[
\text{Recall} = \frac{TP}{TP + FN}
\]
\[
F\text{-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

The results of ten-fold cross-validation indicate that random forest performs best in terms of accuracy, recall, and F-measure with comparable precision, compared with other algorithms with good interpretability, including C4.5, naïve Bayes, and Bayesian networks. More details are shown in Table 3.

The risk planning module is implemented to reduce the risk planning problem into a linear PBO problem, which can be solved by a pseudo-Boolean solver (see Section 4 for more detail).

<table>
<thead>
<tr>
<th>Risk dimension</th>
<th>Abb.</th>
<th>Risk factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organizational environment risk</td>
<td>Org1</td>
<td>Change in organizational management during the project</td>
</tr>
<tr>
<td>Org2</td>
<td>Corporate politics with negative effect on project</td>
<td></td>
</tr>
<tr>
<td>Org3</td>
<td>Unstable organizational environment</td>
<td></td>
</tr>
<tr>
<td>Org4</td>
<td>Organization undergoing restructuring during the project</td>
<td></td>
</tr>
<tr>
<td>User risk</td>
<td>User1</td>
<td>Users resistant to change</td>
</tr>
<tr>
<td>User2</td>
<td>Conflict between users</td>
<td></td>
</tr>
<tr>
<td>User3</td>
<td>Users with negative attitudes toward the project</td>
<td></td>
</tr>
<tr>
<td>User4</td>
<td>Users not committed to the project</td>
<td></td>
</tr>
<tr>
<td>User5</td>
<td>Lack of cooperation from users</td>
<td></td>
</tr>
<tr>
<td>Requirement risk</td>
<td>Req1</td>
<td>Continually changing system</td>
</tr>
<tr>
<td>Req2</td>
<td>System requirements not adequately identified</td>
<td></td>
</tr>
<tr>
<td>Req3</td>
<td>Unclear system requirements</td>
<td></td>
</tr>
<tr>
<td>Req4</td>
<td>Incorrect system requirements</td>
<td></td>
</tr>
<tr>
<td>Project complexity risk</td>
<td>Comp1</td>
<td>Project involves the use of new technology</td>
</tr>
<tr>
<td>Comp2</td>
<td>High level of technical complexity</td>
<td></td>
</tr>
<tr>
<td>Comp3</td>
<td>Immature technology</td>
<td></td>
</tr>
<tr>
<td>Comp4</td>
<td>Project involves the use of technology that has not been used in prior projects</td>
<td></td>
</tr>
<tr>
<td>Planning and control risk</td>
<td>PaC1</td>
<td>Lack of an effective project management methodology</td>
</tr>
<tr>
<td>PaC2</td>
<td>Project progress not monitored closely enough</td>
<td></td>
</tr>
<tr>
<td>PaC3</td>
<td>Inadequate estimation of required resources</td>
<td></td>
</tr>
<tr>
<td>PaC4</td>
<td>Poor planning</td>
<td></td>
</tr>
<tr>
<td>PaC5</td>
<td>Project milestones not clearly defined</td>
<td></td>
</tr>
<tr>
<td>PaC6</td>
<td>Inexperienced project manager</td>
<td></td>
</tr>
<tr>
<td>PaC7</td>
<td>Ineffective communication</td>
<td></td>
</tr>
<tr>
<td>Team risk</td>
<td>Team1</td>
<td>Inadequately trained development team members</td>
</tr>
<tr>
<td>Team2</td>
<td>Inexperienced team members</td>
<td></td>
</tr>
<tr>
<td>Team3</td>
<td>Team members lack specialized skills required by the project</td>
<td></td>
</tr>
</tbody>
</table>
6. Application of IF-ISPRP

To demonstrate the integrative risk analysis and planning, we applied the model to a real social media platform project called “Guangzhou Wireless City.” The project aims to establish an integrated social media platform, which provides a variety of social media applications for people in a city covered with ad-hoc wireless network. Due to privacy concern, basic information of the company is not presented in the paper.

A variety of information is required as input parameters, including the state assessment of risk factors, the many-to-many relationships between risk-control actions and risk factors, and the execution cost of each risk-control action.

The assessment of each input parameter was conducted by an expert panel, which is composed of three project key stakeholders, including the project manager, chief technical officer, and customer manager. There are three rounds for the panelists to reach a consensus. In the first round, each participant anonymously and independently gives his or her own opinion and a corresponding reason for each parameter. In the second round, based on the other experts’ opinions and reasons collected in the first round, each participant is asked to revise his or her opinion. In the third round, the differences in opinions are discussed until consensus is reached over all parameters.

In the present application, the reached consensus is shown in Table 4. The state of each risk factor is either high or low, indicating the degree of risk. Not surprisingly, the social software project mainly suffers from requirement and user risks. For example, conflicts exist among users in terms of software functions, due to distinct purposes of using social software (“User2 Conflict between users”). In addition, user requirements are vague and unstable (“Req1 Continually changing system” and “Req3 Unclear system requirements”). Taking the states of risk factors as input, the risk analysis module predicted the “Guangzhou Wireless City” project as a failure. The prediction implied that it is necessary to carry out the subsequent planning process.

First, the expert panel was asked to choose a subset among the 74 risk-control actions listed by Benaroch et al. [5] according to the real situation of the company. The final consensus covers 12 risk-control actions as shown in Table 5.

Second, the expert panel was also asked to arrive at a consensus on how each risk factor can be improved by specific risk-control actions given a specific execution cost. Considerable amount of effort was spent to confirm this information (shown in Table 6). Each action can change the state of the corresponding risk factor from high to low with a given execution cost.

Given the above parameters as input, a cost-minimal action set for risk control was computed by the risk planning module as shown in Table 7.

The solution shows that if two actions, Act3 and Act12, are applied with a total execution cost of 90,000 RMB, the “Guangzhou Wireless City” project will be predicted as a success. In total, the risk analysis model generates ten decision trees. Fig. 4 only shows how the Act3 changes the states of risk factors for achieving a success prediction. The effect of Act12 is manifested in another decision tree(s). As the state of PaC4 is originally high, the project is consequently predicted as a failure in Fig. 4 (node 4). The state of PaC4 can be changed into low by Act3 “Research/better project planning,” with an execution cost of 40,000 RMB; the project then changes from a failure prediction to as a success prediction (Org2 = low, Team2 = low, PaC7 = low, so falls into node 11 in Fig. 4).

It is interesting that the model chooses Act3 to make the project outcome prediction a success rather than take actions Act6, Act7 or Act8, which aim to mitigate requirement risks (as shown in Table 6). A rational explanation may be that such social software project has inherent risk of frequent requirement changes, thus improving project planning ability is more effective than directly mitigating the requirement risks.

7. Discussion

The proposed framework, IF-ISPRP, can adopt various risk factor models, thus it can be adapted to other risk management fields. Moreover, we discuss the established model in the following aspects:

1) Objective function. The model minimizes the total execution cost for risk control, i.e., generate a cost-minimal action set. However, it can be adapted to optimize alternative objective functions, such as the total utility or execution time. The model outputs only one solution with the minimal total cost. However, it can be adapted to produce multiple solutions that are equally good; therefore, project stakeholders are given the freedom to choose the one best suited to their situations.

2) Method. The MMAKD method uses random forest as the risk analysis model. Random forest can be transformed to a set of (classification) rules with a good interpretability, and is well-fitted for solving and explaining risk planning problems. Bayesian network may be another good choice for the risk analysis model because it has also a good interpretability and can cope with uncertainty. However, we need to extract rules from a Bayesian network for applying the MMAKD method. Although there exist methods for extracting rules from a Bayesian network [13], the extraction of rules is clearly out of the scope of this study; thus Bayesian network is not used as the risk analysis model in this study.

Table 3
Comparison of performances.

<table>
<thead>
<tr>
<th>Random forest</th>
<th>C4.5</th>
<th>Naïve Bayes</th>
<th>General Bayesian networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>82.2%</td>
<td>76.2%</td>
<td>76.9%</td>
</tr>
<tr>
<td>Precision</td>
<td>85.9%</td>
<td>84.0%</td>
<td>93.5%</td>
</tr>
<tr>
<td>Recall</td>
<td>50.3%</td>
<td>83.2%</td>
<td>73.3%</td>
</tr>
<tr>
<td>F-Measure</td>
<td>0.881</td>
<td>0.836</td>
<td>0.823</td>
</tr>
</tbody>
</table>

PS.: High risk project is treated as positive class.

Table 4
Risk assessment of the “Guangzhou Wireless City” project.

<table>
<thead>
<tr>
<th>Risk</th>
<th>State</th>
<th>Risk</th>
<th>State</th>
<th>Risk</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Org1</td>
<td>High</td>
<td>Req1</td>
<td>High</td>
<td>PaC2</td>
<td>Low</td>
</tr>
<tr>
<td>Org2</td>
<td>Low</td>
<td>Req2</td>
<td>High</td>
<td>PaC3</td>
<td>Low</td>
</tr>
<tr>
<td>Org3</td>
<td>Low</td>
<td>Req1</td>
<td>High</td>
<td>PaC4</td>
<td>High</td>
</tr>
<tr>
<td>Org4</td>
<td>Low</td>
<td>Req4</td>
<td>Low</td>
<td>PaC5</td>
<td>High</td>
</tr>
<tr>
<td>User1</td>
<td>Low</td>
<td>Comp1</td>
<td>Low</td>
<td>PaC6</td>
<td>Low</td>
</tr>
<tr>
<td>User2</td>
<td>High</td>
<td>Comp2</td>
<td>High</td>
<td>PaC7</td>
<td>Low</td>
</tr>
<tr>
<td>User3</td>
<td>Low</td>
<td>Comp3</td>
<td>Low</td>
<td>Team1</td>
<td>High</td>
</tr>
<tr>
<td>User4</td>
<td>High</td>
<td>Comp4</td>
<td>Low</td>
<td>Team2</td>
<td>Low</td>
</tr>
<tr>
<td>User5</td>
<td>High</td>
<td>PaC1</td>
<td>Low</td>
<td>Team3</td>
<td>Low</td>
</tr>
</tbody>
</table>

Table 5
Candidate risk-control actions.

<table>
<thead>
<tr>
<th>ID</th>
<th>Risk control action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Act1</td>
<td>Get corporate sponsor</td>
</tr>
<tr>
<td>Act2</td>
<td>Plan for change management</td>
</tr>
<tr>
<td>Act3</td>
<td>Research/better project planning</td>
</tr>
<tr>
<td>Act4</td>
<td>Put QA process in place</td>
</tr>
<tr>
<td>Act5</td>
<td>Deployed in a small market segment</td>
</tr>
<tr>
<td>Act6</td>
<td>More testing of IT elements of the project</td>
</tr>
<tr>
<td>Act7</td>
<td>5–10% of investment gone into research prior to full commitment</td>
</tr>
<tr>
<td>Act8</td>
<td>Staged implementation-rolling releases</td>
</tr>
<tr>
<td>Act9</td>
<td>MOSCOW (get project’s Must haves, then Should haves, Could haves, and finally Would haves)</td>
</tr>
<tr>
<td>Act10</td>
<td>Contingency plan/possibility to abandon existed</td>
</tr>
<tr>
<td>Act11</td>
<td>Get corporate sponsor</td>
</tr>
<tr>
<td>Act12</td>
<td>Extra training of employees</td>
</tr>
</tbody>
</table>
3) Continuous risk management. It is usually impossible to eliminate all risks by a single round of risk analysis and planning. A long-term, continuous, and iterative process of risk assessment, analysis, planning, resolution, and tracking, needs to be performed on a regular interval (weekly, monthly, at every milestone, etc.) throughout the lifecycle of a software project. During and after the implementation of the risk-control plan, the states of the risk factors should be dynamically monitored and tracked. The feedback information should be provided for subsequent reassessment of risk factors and for making a new risk-control plan (using the proposed framework).

8. Conclusion and limitations

Risk analysis and planning are complex, resulting in difficulties to manage risks effectively through subjective judgment. So far, the SPRM domain lacks an empirical integrative intelligent model for risk analysis and planning. The current study has two major contributions:

1) The proposed IF-ISPRP is the first integrative framework for intelligent SPRM, which aims at generating a cost-minimal risk-control action set. In addition, an empirical model is established based on real software project data.

2) The proposed MMAKD method considers the many-to-many relationship between risk factors and risk-control actions, which is prevalent in practical risk planning. Compared with existing studies that use integer-programming technology [6], this method directly uses a risk analysis model and is more reasonable.

To demonstrate the effectiveness of the model, application of the model on a social software project showed how the model generates a risk-control plan to mitigate the whole project risk. To control the high-level requirement and user risks, the model suggests the need to improve project planning ability and train the development team. This suggestion may also be applied to other social software projects.

However, our research has the following limitations:

1) Although the MMAKD method can find a cost-minimal risk-control action set, it does not consider the execution order of risk-control actions. However, it is very hard or even unrealistic for practitioners to define complete information for order constraints among different candidate risk-control actions. Therefore, our model assumes that all actions can be executed simultaneously.

2) In order to facilitate the modeling and application process, three reasonable assumptions are taken to simplify the risk planning problems. For example, the assumption that the effect of an action is defined using a monotonically increasing step function on the execution cost of the action (i.e., the more costs we input, the better effectiveness of risk control we will have) only works in certain situations. Hence, if extreme result occurs, the experts must make additional judgments and adjustments before the implementation of the risk-control plan.

As George Box said, “All models are wrong, but some models are useful” [10, p. 202], the established model is only a simplification of the real world it represents and as such, it is merely suitable for a limited range of situations.

Acknowledgments

This research was partly supported by the National Natural Science Foundation of China (71271061, 70801020), Science and Technology Planning Project of Guangdong Province, China (2010B010600034, 2012B091100192), and Business Intelligence Key Team of Guangdong University of Foreign Studies (TD1202).

Table 6

<table>
<thead>
<tr>
<th>Org1</th>
<th>Org2</th>
<th>Org3</th>
<th>User1</th>
<th>User2</th>
<th>User3</th>
<th>User4</th>
<th>Req1</th>
<th>Req2</th>
<th>Req3</th>
<th>Req4</th>
<th>Comp2</th>
<th>PaC1</th>
<th>PaC4</th>
<th>PaC5</th>
<th>Team1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Act1</td>
<td>15W</td>
<td>Act2</td>
<td>10W</td>
<td>Act3</td>
<td>3W</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Act4</td>
<td>20W</td>
<td>Act5</td>
<td>5W</td>
<td>Act6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Act7</td>
<td></td>
<td>Act8</td>
<td>35W</td>
<td>Act9</td>
<td>10W</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Act10</td>
<td>50W</td>
<td>Act11</td>
<td>5W</td>
<td>Act12</td>
<td>5W</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

W stands for ten thousand CNY.

Table 7

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
<th>Cost</th>
<th>Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Act3</td>
<td>4</td>
<td>Change PaC4 from high to low</td>
</tr>
<tr>
<td>2</td>
<td>Act12</td>
<td>5</td>
<td>Change Team1 from high to low</td>
</tr>
</tbody>
</table>

Fig. 4. Effect of risk-control action.
References


Yong Hu is currently an Associate Professor and Chair in the Department of e-commerce, and Director of Business Intelligence and Knowledge Discovery at the Guangdong University of Foreign Studies and Sun Yat-Sen University. He received his BSc in Computer Science, MPhil and PhD. in Management Information Systems from Sun Yat-Sen University. His research interests are in the areas of business intelligence, software project risk management, e-commerce and decision support systems. He has published in a number of journals and conferences such as DSS, ESWA and IEEE ICDM. Dr. Hu’s research is supported by the National Natural Science Foundation of China, the Science and Technology Planning Project of Guangdong Province, and “211 Project” of the Guangdong University of Foreign Studies.

Jianfei Du is currently a lecturer in Institute of Business Intelligence and Knowledge Discovery in Guangdong University of Foreign Studies. He received the Ph.D. degree from the State Key Laboratory of computer science, Institute of Software, Chinese Academy of Sciences, and both the Master degree and the Bachelor degree from Sun Yat-Sen University in PR China. His current research interests include knowledge representa- tion and reasoning, and semantic web. He has published papers in major conferences in these areas, such as IJCAI, AAAI, UAI, WWW and ISWC.

Xiangzhou Zhang is a Ph.D. student in Sun Yat-Sen University and working as an assistant researcher in Institute of Business Intelligence and Knowledge Discovery in Guangdong University of Foreign Studies and Sun Yat-Sen University. He has received his B.S. degree in Computer Science from Sun Yat-Sen University, and M.S. degree in Management from Guangdong University of Foreign Studies. His research interests include data mining, software project risk management, and business intelligence.
Xiaoling Hao is an associate professor at the School of Information Management and Engineering, Shanghai University of Finance and Economics. She was a visiting scholar at the Foster School of Business, University of Washington. Her research interests include project management and IT governance.

Prof. Eric Ngai is a Professor in the Department of Management and Marketing at The Hong Kong Polytechnic University. His current research interests are in the areas of e-commerce, Supply Chain Management, Decision Support Systems and RFID Technology and Applications. He has published papers in a number of international journals including MIS Quarterly, Journal of Operations Management, Decision Support Systems, IEEE Transactions on Systems, Man and Cybernetics, Information & Management, Production & Operations Management, and others. He is an Associate Editor of European Journal of Information Systems and serves on editorial board of six international journals. Prof. Ngai has attained an h-index of 13, and received 510 citations, ISI Web of Science.

Ming Fan is an associate professor in information systems at Foster School of Business, University of Washington. He received his B.S. from Nanjing University, China and Ph.D. degree in information systems management from the business school in University of Texas at Austin. He is an associate editor of Information Systems Research, Decision Sciences Journal, and Decision Support Systems.

Mei Liu is currently an Assistant Professor in the Department of Computer Science at New Jersey Institute of Technology. She received her Ph.D. degree in Computer Science from the University of Kansas, Lawrence, USA and completed her postdoctoral training as an NIH-NLM research fellow in the Department of Biomedical Informatics at Vanderbilt University, Nashville, USA. Her research interest includes data mining, machine learning, text mining, decision support systems, quantitative investment, and medical informatics. She has published a number of papers in conferences and journals such as Bioinformatics, JAMIA, ESWA, EURASIP Journal on Applied Signal Processing, BMC Bioinformatics, PLoS ONE, and IEEE ICDM.