2010 3rd International Conference on Information Management, Innovation Management and Industrial Engineering

A Unified Intelligent Model for Software Project Risk Analysis and Planning

Yong Hu, Xiangzhou Zhang, Xin Sun, Jing Zhang, Jianfeng Du, Junkai Zhao
Business Intelligence and Knowledge Discovery, School of Business
Guangdong University of Foreign Studies, Sun Yat-sen University
Guangzhou, China
e-mail: zhxzhou86@foxmail.com

***Abstract*—Software project development still faces high failure rate. Nowadays, researches on software project risk majorly remain in risk theoretical model, risk identification and risk analysis, lacking empirical models which unify risk analysis and risk planning procedures. This research firstly collected real data from mainland China, and used Classification Association Rules (CARs) to provide a highly operational and planning-oriented software project risk management (SPRM) framework. Then we code programing problem, i.e. risk planning, into Constraint Satisfaction Problem (CSP), to establish a Unified Intelligent Model for Risk Analysis and Planning (UIM-SPRAP). Thus we provide a new solution to establish a framework unifying risk analysis and planning. UIM-SPRAP is the earliest model of this field; can be used to provide intelligent decision support to SPRM.**

Keywords-Software Project Development; Unified Intelligent Risk Analysis and Planning Model; Classification Association Rules; Constraint Satisfaction Problem

I. Introduction

So far, software development has been of high risk. Standish Group surveyed on more than one hundred thousand software projects, which demonstrated that from 1994 to 2009, success rate had climbed up, but still below 1/3 (16% and 32% respectively) [1].

Risk management is one of the most important management works of software development. Charatte regarded that management of large software project is risk management [2]. IEEE research revealed that in software system, 50%-70% of risks could be detected, 90% could be avoided, and risk management had a leverage effect with ROI about 700%-2000% [3].

Boehm thought that software project risk management (SPRM) was trying to use feasible principles or practices to regularly control the risk which could affect project success; its goal is to identify, depict and eliminate risk factors, lest threatening successful operation of software [4].

Our research aim and contribution are as follows:

National and international SPRM researches mostly remain on risk theoretical model, risk identification and risk analysis. No SPRM frameworks provide unified intelligent decision support form risk analysis to risk planning, so it's hard to manage software project risk effectively. In terms of our collected papers, unified risk analysis and planning exploratory models based on data mining methods are still in the initial stage. Our research is the first to build up a unified risk analysis and planning exploratory model based on Classification Association Rules (CARs). We used real software development data collected in mainland China, so the proposed UIM-SPRAP trained by these data will be more pertinent in mainland China SPRM theoretical research and practice.

II. Methodology

In this section, we introduce our proposed framework, in which only the risk mitigation strategies and risk planning parts contain detailed description, because we follow common ways of risk identification and analysis.

A. Planning-oriented SPRM framework

Our proposed planning-oriented SPRM framework has three main phases, as shown in Fig. 1. Risk identification lies in infrastructure; based on the identification result, risk analysis is executed; if the result of risk analysis predicts that the project will fail, risk planning is conducted. Risk planning is the key to risk mitigation, which is the ultimate goal of SPRM.

Risk database is used to document and provide software project risk information, including risk factor database, risk factor states database, risk mitigation strategies database, successful project database, failed project database, etc. Mature software organization do very focus on risk information collection, analysis and process, which are the foundation and basis of risk management. So far, in many software process improvement models, process database (including risk database) is one of the important marks of evaluating organization process improvement ability.

B. Risk Identification

Risk identification is to identify and record the sources of negative project effect (i.e. risk factors), and transfer the uncertainties and potential problems in identification project into tangible risk factors that could be described and measured. Our research adopts the six dimension theoretical model of Wallace and Keil [5] to be the infrastructure of risk identification.

978-0-7695-4279-9/10 $26.00 © 2010 IEEE
DOI 10.1109/ICIII.2010.504

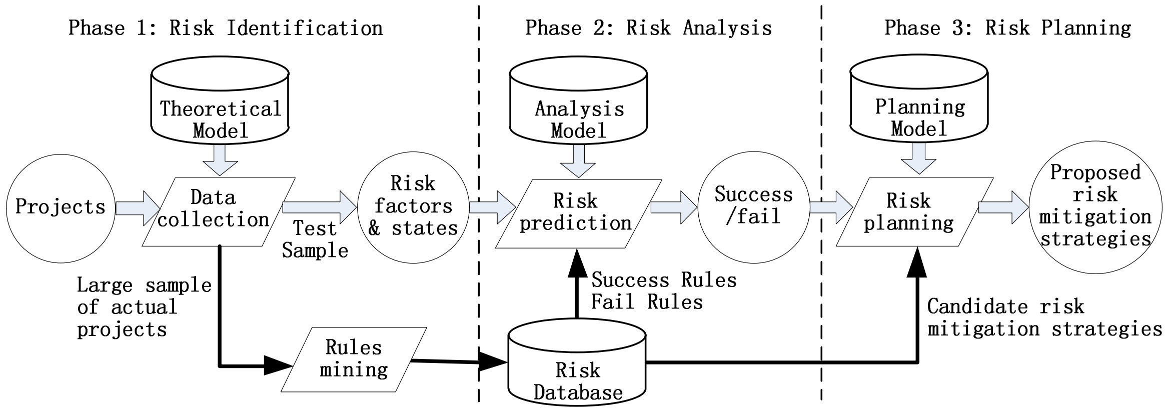

Figure 1. Planning-Oriented SPRM Framework

C. *Risk Analysis*

Risk analysis is the process of converting risk data into risk mitigation information, and assessing project risk level.

Project assessment usually adopts classification method to classify projects into successful projects (low risk) or failed projects (high risk). Common techniques include Decision Tree, Bayesian Network, Neural Network, CARs, K-Nearest Neighbors method, etc. Our research applies CARs, which is also called Predictive Association Rules, which directs at association rule used to distinguish or predict instance class labels [6]. CARs discovery usually includes two steps: (1)Distinguish association rules which have right hand side class labels, as "($factor_1$, $state_1$),…,($factor_n$, $state_n$) → *project succeed/fail*"; (2)From the already discovered rules, choose the prior rules to cover the training sets according to evaluation standards such as *confidence*, *support* and *rule length*. Megiddo et al. used statistical significance test to investigate the result of association rules mining and found that less than 0.1% were "fake discovery" [7]. CARs technology is widely used in SPRM field now [8].

We modified the typical Apriori algorithm to construct risk analysis model, which only generates generic project success/failure rules, that is to say any subset of antecedent of success/failure rule is not success/failure rule, in order to reduce rule amounts, as shown in Fig. 2.

D. *Risk Mitigation Strategy*

Our research collects and induces a set of risk mitigation strategies through paper reviews and experts interviews [4],[9],[10], [11] [12], [13], [14]], and they could be classified into five dimensions, as shown in Table I.

Each risk mitigation strategy could control several risk factors and have different cost, and different cost could reduce risk to different levels, thus we should considerate more on the constraints when making a specific risk planning.

E. *Risk Planning*

Risk planning is, based on the risk analysis result, to apply various auxiliary decision technologies to achieve risk contingency plan, risk avoidance methods, risk mitigation strategies and action plan, etc.

Due to the strong coupling of risk analysis and risk planning, consistent representation (data structure) is more convenient, thus the CARs pattern we adopted is well suitable.

Risk planning is such a procedure:

- *Input*: risk mitigation strategy database (including strategy name, strategy cost, and changes of risk factors states) and risk factors database (including risk factor names and states)
- *Output*: a set of proposed risk mitigation strategies (a subset of risk mitigation strategy database)
- *Constraint*: satisfy the risk planning goals defined in SEI CRM model [15]:

1) Avoids factor states leading to project failure;

2) Reaches factor states leading to project success;

3) Satisfies the above conditions and minimize total strategies cost.

The task of risk planning can be regarded as such a simple process: find out a set of risk mitigation strategies, through which we can change project risk factors states and then convert a "fail" project, i.e. "($factor_1$, $state_1$),…,($factor_n$, $state_n$) → *project fail*", into a "success" project, i.e. "($factor'_1$, $state'_1$),…,($factor'_n$, $state'_n$) → *project succeed*".

In this research, the solution of risk mitigation strategies set is an integer programming problem; it can be resolved by using common integer programming tools, but the result is hard to interpret, obstructing software project manager to accurately understand the risk planning. One of the common technologies of programming problem is coding it as a Constraint Satisfaction Problem (CSP). CSP includes Integer programing problem, but it's more interpretable and effective [16]. Furthermore, CSP is often coded as Satisfiability Problem (SAT).

So we code the task of risk planning as SAT with linear optimization objective and invoked a Pseudo-Boolean SAT Solver (PB-Solver) *minisat+* to make a solution. The coding task is to define constraint and objective function, which are as follows:

1) Possible initial states of each risk factor;

2) Mutual exclusivity within each risk factor's different initial states;

3) Possible final states of each risk factor;

4) Mutual exclusivity within each risk factor's different final states;

5) Risk factor final state is of lower risk than its initial state;

6) Mutual exclusivity within each risk mitigation strategy's different costs;

7) Effect of each risk mitigation strategy on various risk factors' states;

8) Initial state of each risk factor of the test sample;

9) Final states of the test sample meet at least one success rule;

10) Final states of the test sample meet no fail rule;

11) Linear optimization objective is searching for risk mitigation strategies set with minimized total cost.

III. EXPERIMENT AND RESULT

A sound risk theoretical model is the key to precisely predict risk outcomes. Carr's model [17], developed by CMU/SEI, is very complex, which consists of 194 risk factors and many detailed information, so that it needs lots of project data to reach high statistical validity, which is especially hard to apply to middle and small sized project in China. Our research adopts Wallace and Keil's model [5] that has 27 risk factors to form our investigation questionnaire (each factor has 5 possible states/risk level, i.e., *low*, *a bit low*, *moderate*, *a bit high*, *high*).

500 questionnaires were delivered and 302 returned were valid, among them 226 were about failed projects, 76 about successful projects. The successful rate is about 25% （76/302） which is in line with the reality of China software project development. The respondents mostly have enriched software project development experience, 83% of which have more than 3 year experience and 46% have taken up management position in the projects, mainly department manager, project manager or project technological guidance. And more than half of the projects were finished within the latest 3 years; hence the collected sample can well reflect recent domestic SPRM reality.

We set *support* to be 3 and the *confidence* to be 95%, we totally find 10 success rules and 3130 fail rules, Table II only presents the success rules.

In one of surveys, we suppose a risk mitigation strategy database which contains 530 cases of different cost and utilities. The survey initially meets no success rule and 715 fail rules. We then used *minisat+* to get a resolution which contains 9 feasible strategies (worth RMB 75,000) to make the project meets 7 success rules and 0 fail rules. Partial strategies are as follows: launch RMB 9,000 to better internal training, in order to lower "Inadequately trained development team member" risk from *high* to *low*; launch RMB 15,000 to hire experts or consultancies to lower "Inexperienced project manager" risk from *moderate* to *low*, to lower "Project involved the use of new technology" risk from *high* to *a bit high*; launch RMB 7,000 to make the requirement analysis process more normalized lower "Unclear system requirements" risk from *moderate* to *low*; etc.

TABLE I. COMMON SPRM RISK MITIGATION STRATEGIES

Team management	**Time management**	**Third party support**
▪ Build elite team ▪ Employ experienced experts ▪ Institutionalized communication ▪ Internal training …	▪ Formulate more detailed quality standard ▪ Set up clear milestone ▪ Rewrite partial project planning/ further planning …	▪ Hire experts, consultants or consultancies ▪ External training ▪ Outsourcing ▪ Hire external staffs …
User management	**Project management**	
▪ Obtain mor user involvement/ user support ▪ Strengthen communication with user to improve user understanding ▪ Compose detailed user manual …	▪ Employ effective requirement management method ▪ Normalize and streamline requirement analysis process ▪ Rank requirement according to importance ▪ Use rapid prototyping/ incremental development/ evolution development/ modular design …	

TABLE II. DISCOVERED SUCCESS RULES

ID	Risk factors states
1	Org2 =*low*, P&C4 =*low*, P&C5=*low*
2	Org2=*low*, P&C5=*low*, P&C6=*low*
3	Org2=*low*, P&C5=*low*, Team3=*low*
4	User1=*low*, Req3=*low*, Team1=*a bit low*
5	User2=*low*, P&C3=*low*, Team1=*a bit low*
6	User5=*low*, P&C1=*low*, P&C5=*low*
7	User5=*low*, P&C5=*low*, Team3=*low*
8	Req3=*low*, P&C5=*low*, Team1=*a bit low*
9	P&C5=*low*, P&C6=*low*, Team2=*a bit low*
10	Org2=*low*, User1=*low*, User2=*low*, User4=*low*

User1: Users resistant to change
User2: Conflict between users
User4: Users not committed to the project
User5: Lack of cooperation from users
Team1: Inadequately trained development team members
Team2: Inexperienced team members
Team3: Team members lack specialized skills required by the project
Org2: Corporate politics with negative effect on project
Req3: Unclear system requirements
P&C1: Lack of an effective project management methodology
P&C3: Inadequate estimation of required resources
P&C4: Poor project planning
P&C5: Project milestones not clearly defined
P&C6: Inexperienced project manager

```
Ck : candidate frequent k-itemsets
Lk : frequent k-itemsets

Input:   D: Transaction DB;
         min_sup: minimal support count threshold;
         min_conf: minimal confidence threshold
Output: L: frequent itemsets in D

(1)  L1 = find_frequent_1-itemsets(D);
(2)  for (k = 2; Lk-1 !=∅; k++)
(3)     Ck = aproiri_gen(Lk-1, min_sup);
(4)     foreach tran t ∈ D
(5)        Ct =subset(Ck, t.itemset); //subsets of t.itemset in Ck
(6)        foreach candidate c∈Ct
(7)           c.sup++;
(8)           if t.bSuccess then c.s_sup++; else c.f_sup++;
(9)     foreach candidate c ∈ Ck
(10)       if !c.bSubsetSRule then
(11)          c.bSubsetSRule = (c.s_sup / c.sup ≥ min_conf);
(12)          c.bMostGeneralSRule = true;
(13)       else c.bMostGeneralSRule = false;
(14)       if !c.bSubsetFRule then
(15)          c.bSubsetFRule = (c.f_sup / c.sup ≥ min_conf);
(16)          c.bMostGeneralFRule = true;
(17)       else c.bMostGeneralFRule = false;
(18)  Lk={c∈Ck | c.sup ≥ min_sup,
           c.bMostGeneralSRule || c.bMostGeneralFRule =true};
(19)  ruturn L=∪Lk (k = 1, 2, 3...);
```

```
procedure find_frequent_1-itemsets(D)
(1)  foreach item i ∈ D  set Ci.itemset = {{i}};
(2)  foreach tran t ∈ D  update all Ci.sup, Ci.s_sup, Ci.f_sup;
(3)  foreach Ci
(4)     if Ci.s_sup ≥ min_sup then set Ci.bSubsetSRule,
                                  Ci.bMostGeneralSRule to be true
(5)     if Ci.f_sup ≥ min_sup thenset Ci.bSubsetFRule,
                                  Ci.bMostGeneralFRule to be true
(6)  return L1= {Ci | Ci.sup ≥ min_sup};

procedure apriori_gen(Lk-1)
(1)  foreach pair of l1, l2 ∈ Lk−1
(2)     if (l1.itemset[1]=l2.itemset[1]) ∧ …∧ l1[k-2]=l2[k-2])∧(l1[k-1]<l2[k-1]) then
            set c.itemset={l1.itemset[1],…, l1.itemset[k-1], l2.itemset[k-1]};
(3)     if prune(c, Lk-1) then delete c; else add c to Ck;
(4)  return Ck;

procedure prune(c, Lk-1)
(1)  c.bSubsetSRule = c.bSubsetFRule = false;
(2)  foreach (k-1)-itemset s ∈ c.itemset
(3)     if there is no d ∈ Lk-1 satisfies d.itemset=s then return true;
(4)     Let d to be the element of Lk-1 that satisfies d.itemset=s;
(5)     if d.bSubsetSRule then c.bSubsetSRule = true;
(6)     if d.bSubsetFRule then c.bSubsetFRule = true;
(7)     if c.bSubsetSRule && c.bSubsetFRule then return true;
(8)  return false;
```

Figure 2. Modified Apriori algorithm for rules mining

IV. Conclusion

Our research proposed a highly operational and planning-oriented SPRM framework based on CARs. Besides we code the risk planning into CSP, and finally into SAT, which can be solved by using *minisat+*. Our methodology presents new resolution guidance to the research of unified model of software project risk analysis and planning. And our UIM-SPRAP is the first unified software project risk analysis and planning model. We use real software development project data of China to test UIM-SPRAP, and provide intelligent decision support to the SPRM from risk analysis to planning.

Acknowledgment

This research was partly supported by National Natural Science Foundation of China (NSFC, Project No.: 70801020), Science and Technology Planning Project of Guangdong Province (Project No.: 2010B010600034), key program of National Social Science Foundation of China (NSSFC, Project No.: 08AJY038), and the third-stage key subjects construction project of 211 Project of Sun Yat-sen University and 211 Project of Guangdong University of Foreign Studies.

References

[1] The Standish Group. 2009 Standish Chaos Report. The Standish Group International, Inc., 2009.

[2] R.N. Charette, “Large-scale project management is risk management”, IEEE Software, vol. 13, Jul. 1996, pp. 110-117, doi: 10.1109/52.526838.

[3] T. Lister, “Interview with Tim Lister”, IEEE Software, vol. 14, 1997, pp. 18-19.

[4] B.W. Boehm, “Software risk management: principles and practices”, IEEE Software, vol. 8, Jan/Feb. 1991, pp. 32-41, doi: 10.1109/52.62930.

[5] L. Wallace, M. Keil, and A. Rai, “Understanding software project risk: a cluster analysis”, Information and Management, vol. 42, Dec. 2004, pp. 115-125, doi: 10.1016/j.im.2003.12.007.

[6] B. Liu, W. Hsu, and Y. Ma, “Integrating classification and association rule mining”, Proc. Knowledge Discovery and Data Mining (KDD 98), New York, Aug. 1998, pp. 80-86, doi: 10.1.1.48.8380.

[7] N. Megiddo, and R. Srikant, “Discovering predictive association rules”, Proc. Knowledge Discovery and Data Mining (KDD 98), New York, Aug. 1998, pp. 274-278, doi:10.1.1.42.5517.

[8] M.N.M. Garcia, I.R. Román, F.J.G. Peñalvo, and M.T. Bonilla, “An association rule mining method for estimating the impact of project management policies on software quality, development time and effort”, Expert Systems with Applications, vol. 34, Jan. 2008, pp. 522-529, doi:10.1016/j.eswa.2006.09.022.

[9] H.J. Leavitt. “Applied organization change in industry structural, technical and human approaches”, New Perspectives in Organizational Research, New York, John Wiley, 1964, pp. 55–70.

[10] M. Keil, P.E. Cule, K. Lyytinen and R.C. Schmidt, “A framework for identifying software project risks”, Communications of the ACM, vol. 41, Nov. 1998, pp. 76-83, doi:10.1145/287831.287843.

[11] M. Benaroch, Y. Lichtenstein, and K. Robinson, “Real options in it risk management: an empirical validation of risk-option relationships”, MIS Quarterly, vol. 30, Feb. 2006, pp. 827-864, doi:10.1.1.85.7139.

[12] C.G. Pan, and Y.W. Chen, “An optimization model of CMMI-based software project risk response planning”, International Journal of Applied Mathematics and Computer Science, vol. 1, Jan. 2005, pp. 79-82.

[13] A. Hakim, and H. Hakim. “A practical model on controlling the ERP implementation risks”, Information Systems, vol. 35, Apr. 2010, pp. 204-214, doi:10.1016/j.is.2009.06.002.

[14] J.Y.Li, O. P. N. Slyngstad, M. Morisio, “A State-of-the-Practice Survey of Risk Management in Development with Off-the-Shelf Software Components”, Software Engineering, IEEE Transactions on Software Engineering, vol. 34, Mar. 2008, pp. 271-286, doi: 10.1109/TSE.2008.14.

[15] C. Chittister, R. Kirkpatrick and R.L. Van Scoy, “Risk management in practice”, Technical Review SEI-93-TechReview-004, 1993.

[16] H. Kautz, D. McAllester, and B. Selman, “Encoding plans in prepositional logic”, Proc. International Conference on Principles of Knowledge Representation and Reasoning (KR 96), Nov. 1996, pp. 374-384, doi:10.1.1.30.9946.

[17] M.J. Carr, S.L. Konda, I. Monarch, F.C. Ulrich, and C.F. Walker, “Taxonomy-based risk identification”, Technical Report CMU/SEI-93-TR-006.