JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 30, 1279-1301 (2014)

# An Information-Entropy-based Risk Measurement Method of Software Development Project*

RONG JIANG[1,2]
[1]*School of Information*
*Yunnan University of Finance and Economics*
*Kunming, 650221 P.R. China*
[2]*School of Software*
*Yunnan University*
*Kunming, 650091 P.R. China*
*E-mail: okgoodbetterbest@163.com*

Risk is the major reason for the failure of software development projects. The risk measurement can provide the decision-making data for the risk management and control. The traditional risk measurement method usually assesses in terms of the occurrence probability and the loss of the risk factors, which usually are assessed by experts subjectively. Therefore, they are greatly influenced by artificial factors. In this way, it is difficult to realize the objective and effective measurement. This paper proposes one risk measurement method of software development project based on information entropy, which makes up for the shortcomings of the former studies. This method makes use of information entropy to measure the amount of information so as to measure the software development project risk. In this paper, a new risk checklist is given. First of all, it obtains the risk information through the checklist. Then it obtains the information amount of risk factors by aid of information entropy. This paper concludes that the larger the information amount of the risk factors is, the larger the removed uncertainty is, the smaller the information entropy is and the smaller the risk is. In short, there is a positive correlation between risk and entropy. This paper demonstrates the scientificity and rationality of the method theoretically. Compared with other risk measurement methods, this method manifests certain advantage. The case analysis prove the rationality and feasibility of this method.

***Keywords:*** risk measurement, software development project, information entropy, risk checklist, measurement method

## 1. INTRODUCTION

The characteristics of the software projects determine that the risk is one of its inherent attributes. All software development projects are faced with risks [1]. The highly risky feature of software project management determines that risk management is a key process of software project management, and it is an important branch in the research field of software project management. Effective risk control is highly conducive to the success of a software development project [2].

Received October 12, 2013; revised December 23, 2013; accepted February 15, 21014.
Communicated by Jiann-Liang Chen.
* This work was supported by National Natural Science Foundation of China (Grant No. 61263022, 61303234 and 71362016), China Postdoctoral Science Foundation (Grant No. 2012M521722), National Social Science Foundation of China (Grant No. 12XTQ012), Humanities and Social Sciences Youthful Foundation of the Ministry of Education of China (Grant No. 11YJCZH073), Introduction of Talents Project of Science Research Foundation of Yunnan University of Finance and Economics (Grant No. YC2012D07).

The software development project risk are the factors that lead to the schedule delay, budget overspend, part or overall failure of the software development projects. Research shows [3] too many software development projects end in failure. Fully 25 percent of all software projects are cancelled outright. As many as 80 percent of all software projects run over their budgets, with the "average" software project exceeding its budget by 50 percent. It is estimated that tree-fourths of all large systems are "operational failures". Addison [4] advises that only one-sixth of all projects were completed on time and within budget, one-third of all projects were cancelled and over half were considered 'challenged'.

Research shows [5, 6] most projects fail managerially not technologically. Manny software projects fail completely due to inadequate identification, measurement and management of project risks [1]. One survey of KPMG [7] points out that 55% of out-of-control projects had no risk management, 38% of them made some risk managements while half of them had no risk supervision after the project is proceeded. Therefore, risk is the major reason for the failure of software development projects.

In the late 20th century the risk management was introduced to the software field. Wallace *et al.* [8] claim that to reduce the high failure rate of software projects, managers need better methods and tools to assess and manage software project risk. Boehm [5, 6] and Ropponen [9, 10] point that risk management has become a major method of decreasing the failure of software project management. Addison *et al.* [4] consider that formal risk management greatly improves the likelihood of successful project completion, and it reduces the potential negative consequences of those risks that cannot be avoided. Keil, Cule and Lyytinen [11] assert that the high failure rate is due to managers not taking cautious measures to assess and manage the risks involved in software projects. Schmidt *et al.* [3] declare that among advocated methods for improving software project management, the concept of software project risk management has gained prominence. Ropponen *et al.* [9, 10] believe that by including risk management in a project the exposure to software risk can be reduced and can thereby increase software quality and improve software development.

In a word, risk management has been recognized as an efficient means of improving the high failure rate of the software project as well as played an important role in the best practice of the large-scale software project management.

The risk management of software project refers to the management and control of risk factors that may lead to adverse effects on the project, which can depress risks to an acceptable range. Risk measurement is the basis of the risk management and control, which provides objective data for the scientific decision-making of project manager.

Previous literature has stressed the importance of software project risk assessment and control [8]. To date, however, there has been no widely accepted measure method of software project risk. Although the relationship between risk and software project performance has been continuously examined, project managers who have attempted to apply existing knowledge to mitigate risky areas remain confused [2]. The traditional risk measurement method usually assesses in terms of the occurrence probability and the loss of the risk factors, which usually are assessed by experts subjectively. Therefore, they are greatly influenced by artificial factors. In this way, it is difficult to realize the objective and effective measurement.

In view of these reasons, this paper proposes one risk measurement method of the

software development project based on information entropy, which makes up for the disadvantages of the subjective assessment in terms of occurrence probability and impact degree in previous studies. This method demonstrates the scientificity and rationality of the method theoretically and demonstrates the rationality and feasibility of this method through case study.

The rest of this paper is organized as follows. In section 2, related works are summarized. Section 3 introduces the definition of software development project risk. Section 4 proposes a new measurement model of risk based on information entropy. Section 5 presents an improved measurement model based on the model in section 4. Section 6 describes the advantages of the model proposed in this paper compared with other methods. The conclusion is given in section 7.

## 2. RELATED WORK

There are a lot of studies on the risk and the typical risk management theories include Boehm, CRM, SERIM and Riskit *et al.* In 1989, Boehm published a monograph Software Risk Management [12], which is the first book to explore the software risk management. Its core idea is the list of 10 risk factors. Specific to each risk factor, Boehm provides a series of risk management strategies. This idea efficiently concentrates the attention of the management layer on the key factors, for example, the high risk, high weight and influence factors to success, instead of the numerous details with low priority. SEI (Software Engineering Institution at Carnegie Mellon University of USA) develops a CRM (Con-tinuous Risk Management) model. The principle of SEI is to continuously evaluate the possible factors with bad consequences and determine the most urgent risk. SERIM (Software Engineering Risk Model) [13] put forward by Karolak is recommended by IEEE. He claims that SERIM is based on JIT (Just in Time) strategy. This method accesses the risk factors in the software development in multiple views and it can monitor the risk according to the assessment results at any time of the development cycle. Similar to Boehm, SERIM also is subjectivity-based user input of risk occurrence probability and impact degree. Maryland University proposes Riskit [14] method, whose core is to utilize Riskit, the formal tool analysis graph of risk. This analysis graph can explicitly define the different characteristics of risk, which is more formalized than the regular oral discussion.

Besides, there are also many research achievements in this field. Salmeron and Lopez [15] built a dynamic simulation tool that allows ERP managers to foresee the impact of risks on maintenance goals. Tak Wah Kwan *et al.* [1] propose a management methodology to address risk dependency issues. Li *et al.* [16] propose a software process model with risk management and cost control modules to help improve software process risk management, and based on the process model, a measurement model that includes process risk and software trustworthiness metrics is presented. Andrea Herrmann *et al.* [17] explore practical challenges and needs of risk estimations in general and of their method MOQARE specifically. Shatnawi [18] suggests that there is a relationship between risk levels and object-oriented metrics and that risk levels can be used to identify threshold effects. Sharma and Gupta [19] study conducted on 300 software practitioners, aim to identify organisational climate dimensions and risk dimensions in Indian software

industry through factor analysis. Fu *et al.* [20] establish a probabilistic model which is based on design structure matrix (DSM) to evaluate the risk of change propagation from requirements to software architecture, they predict the risk of change propagation in terms of change propagation probability and change impact. Shi *et al.* [21] propose a linear mixed model (LMM) approach to make individual heterogeneity explicit within the conjoint analysis experimental framework. The model is estimated using an established dataset on Japanese software outsourcing risk perceiving, and the estimation results show that individual heterogeneity does exist. Appari and Benaroch [22] present a risk pricing method that estimates two parameters for every individual risk factor: extra cost incurred per unit exposure, and project sensitivity, to that factor. Gülçi˙n Büyüközkan and Da Ruan [23] propose an integrated multi-criteria evaluation methodology for software development experts and managers to better enable them to position their projects in terms of the associated risk. Benlian and Hess [24] analyze the opportunities and risk associated with adopting SaaS as perceived by IT executives at adopter and non-adopter firms. Sun-jen Huang *et al.* [25] propose a fuzzy decision tree approach for embedding risk assessment information into a software cost estimation model.

Schmidt *et al.* [3] investigate the experienced project managers from different countries by aid of Delphi method. They classified them into 14 classes and 53 software development risk items. Wallace *et al.* [8] collect the opinions of 507 managers from different countries and identified 27 software risk as well as develops 6-dimention risk identification model, including users, requirement, project complexity, plan and control, project team and organization environment. This model makes it easier to understand the software project risk and provides convenience for understanding the influences of risk factors on the project implementation. Keil *et al.* [26] investigate 128 software practitioners and the results show that the risk list can help the identification of more risks and its influence on the risk understanding of practitioners is very significant. On the basis of economics view, R. Costal *et al.* [27] introduce one software project risk assessment technology, which enables the managers to estimate the risk occurrence probability.

Since the punishment of Boehm's monograph Software Risk Management, there have been a lot of gratifying achievements in software risk research, which efficiently improve the success rate of software projects. All those theories and methods have their own application fields and pertinence. With their own advantages and disadvantages, it is difficult to compare with each other. Identification and assessment ricks are two basic steps of the software project risk management. The literatures about the risk identification have been various as well as the adoptable methods. However, the studies on how to assess and quantize the risk factors are relatively few. Though there are expert decision, analog simulation and other methods, these methods always are easy to be greatly affected by the assessors' subjectivity. At present, there are following problems. The qualitative analysis is more than quantitative management and the subjective empirical analysis is more than objective measurement.

Through there have been a lot of risk assessment techniques in the software project field, most of them learned from other engineering project risk management technologies and mostly are major in empirical and subjective analysis. Those methods have solved some risk issues to certain level. However, in practice they usually fail to achieve sound effects. For example, risk assessment methods of Boehm and Karolak [12, 13] both are subjectivity-based user input of risk occurrence probability and impact degree.

## 3. DEFINITION OF SOFTWARE DEVELOPMENT PROJECT RISK

At present, there are still a lot of arguments on the strict definition of software project risk. Some define it as the events that may influence the realization of software targets or result in heavy losses. SEI (Software Engineering Institute at Carnegie Mellon University, USA) defines it as the probability of losses. Besides, there are many other definitions. However, it has been generally accepted that there are two properties of software risk, uncertainty and losses. The uncertainty refers to the probable risk, namely, no 100% risk. Losses refers to the malignant consequences or losses due to the occurrence of risk.

This paper defines the software development project risk as the factors that lead to the schedule delay, budget overspend, part or overall failure of the software development projects. It means the uncertainty in terms of users, requirement, technology, staff, process and organization in the software development process, which may lead result in the unexpected consequences, for example, failing to satisfy the demands of software products/service functions, over-budget, schedule delay and project cancel.

## 4. ONE MEASUREMENT MODEL

### 4.1 Theoretical Foundation and Measurement Model of Risk Information Entropy Measurement

Risk is related to both the occurrence probability and the losses (funds, schedule, performance *etc.*), so people usually utilize triple $R = (X, P, L)$ to describe risk, where $X = (x_1, x_2, \ldots, x_n)$ represents risk factor set; $P = (p_1, p_2, \ldots, p_n)$ represents the occurrence probability of risk factors; and $L = (l_1, l_2, \ldots, l_n)$ represents the losses caused by the risk occurrence. Therefore, we can get $Risk = P \times L$. The main shortcoming of this definition lies in the determination of the occurrence probability of risk factors and losses is very difficult [28]. It is inevitable to make some artificial assumptions by aid of expert estimation method.

Shannon [29] defines information as the reduction of uncertainty, namely,

$$Information = UBC - UAC \tag{1}$$

where *UBC* represents the uncertainty before the communication, *UAC* represents the uncertainty after the communication.

If there is no uncertainty after the communication, the second item of Formula (1) turns into 0 and the receiver receives all the information sent by the sender. If there are not any uncertainty be eliminated, the two items on the right side in the formula above are equal and there is no information sent. Under general condition, one communication always eliminates part of the uncertainty rather than all of it. The information received by information sink party shall eliminate part or all of the uncertainty. In other words, it increases the certainty. Now that the information is defined as the uncertainty eliminated in the communication, the information measurement refers to the measurement of this uncertainty. The to-be-eliminated uncertainty means the occurrence randomness of message. We use probability to measure it. Suppose $p$ as the occurrence probability of mes-

sage $A$; and $I$ represents the information amount of the possible message. Shannon defines it as $I = -\log_2 p$. However, in the communication it is more than one message and it may be a message set, notes as $\{A_1, A_2, \ldots, A_n\}$. Its occurrence possibility is notes as $\{p_1, p_2, \ldots, p_n\}$. It is important to know the overall information ability of the possible message set, which can be expressed with overall average information amount $H$. Shannon defines it as,

$$H(p_1, p_2, ..., p_n) = -k\sum_{i=1}^{n} p_i \ln p_i \tag{2}$$

It represents the overall uncertainty degree, $k$ is a constant.

As mentioned before, uncertainty is the fundamental characteristic of risk. The two aspects of risk, the occurrence probability and losses are uncertain. For this reason, we can define risk in terms of uncertainty. Risk is the set of uncertainty. The larger the uncertainty is, the larger the risk is. The more information we can obtain from one project, the more the eliminated uncertainty is and the smaller the risk is. On contrast, the less information we can obtain, the less the eliminated uncertainty is and the larger the risk is. Risk is defined as the set of uncertainty and information entropy can effectively measure the uncertainty, so we can use information entropy to measure the software project development risk.

If project $P$ has $n$ probable risk factors $F = (f_1, f_2, \ldots, f_n)$, namely there are $n$ uncertainty factors. If nothing is known about these $n$ uncertainty factors, there is no information obtained from the risk factors. In this way, the uncertainty is maximum, so the risk is largest. If there is some information known from $f_i$, namely, part of the uncertainty of $f_i$ is eliminated, so the risk will be reduced. Therefore, we can obtain the following information-entropy-based risk measurement method.

**Risk measurement model 1,**

$$Risk = H_i(f_1, f_2, ..., f_n) = -\sum_{i=1}^{n}\left(\frac{1}{\sum_{j=1}^{n} f_j}\ln\left(\frac{1}{\sum_{j=1}^{n} f_j}\right)\right),$$
$$\text{and} \quad f_j = \begin{cases} 0 & \text{If the uncertainty is not eliminated} \\ 1 & \text{If the uncertainty is has been eliminated} \end{cases} \tag{3}$$

where $i = 1, \ldots, n$.

Formula (3) shows the risk value at certain time of software development project after the risk identification.

Before the risk identification, there is nothing known about the project risk factors, namely, no information is obtained; the uncertainty is not eliminated; the information entropy is maximum, so the risk is largest. At this time, the risk value is,

$$Risk = H_0(f_1, f_2, ..., f_n) = -\sum_{i=1}^{n}\left(\frac{1}{\sum_{j=1}^{n} f_j}\ln\left(\frac{1}{\sum_{j=1}^{n} f_j}\right)\right),$$
$$\text{and} \quad f_j = 1 \tag{4}$$

namely $H_0 = -\sum_{i=1}^{n} \frac{1}{n} \ln \frac{1}{n} = \ln n$.

The decreased risk value after the project risk identification and management is as follows.

$$Risk = H_0(f_1, f_2, \ldots, f_n) - H_i(f_1, f_2, \ldots, f_n) \quad (5)$$

where $H_0(f_1, f_2, \ldots, f_n)$ represents the information entropy before project risk identification and $H_i(f_1, f_2, \ldots, f_n)$ represents the information entropy after project risk identification and assessment.

### 4.2 Acquisition Method of Model Data

Section 4.1 provides the information-entropy-based risk measurement method. However, it fails to solve the issue that when the uncertainty is not eliminated and has been eliminated, namely $f_i = 1$ or $f_i = 0$. This can be determined through the risk factor identification.

Risk identification is the basis of risk management and risk measurement is established on the basis of risk identification. The project risk management should firstly identify the existing risk and its task is to classify the uncertain factors threatening the project. The commonly-used identification methods include, risk checklist, brainstorming method, Delphi method, SWOT analysis method, anonymous risk reporting mechanism and others.

Risk checklist is a list of possible problems according to historical experience. It collects and distinguishes the potential risk through questionnaire; compares it with the to-be-developed software project one by one and determines the probable occurrence of risk in the checklist. At present, there have been a lot of studies on the checklist, which provide various ones for the software projects. The common ones are shown in Table 1.

**Table 1. Common risk checklists.**

| Founder | Risk | Channel |
|---|---|---|
| Boehm (1991) [5, 6] | Ten kinds of risk | Investigated by interviewing numerous experienced project managers |
| Ropponen and Lyytinen (2000) [9] | 6 risk items | Investigated by interviewing 83 project managers from nearly1,100 projects |
| Han and Huang (2007) [30] | 6D risk containing 27 risk factors | Analyzed 115 software projects |
| Wallace *et al.* (2004) [8] | 6D, 27 kinds of risk factors | Investigated the opinions of 507 project managers from different countries |
| Schmidt *et al.* (2001) [3] | 14 kinds and 53 risk items | Investigation on the experienced project managers from different countries by aid of Delphi method |
| Barki *et al.* (1993) [31] | 5 kinds and 23 risk items | Systematic theoretical research overview |
| Heemstra and Kusters (1996) [32] | 9 kinds and 36 risk items | Combination of theory with practice |

Research on theories and practices indicate that risk checklist is a quick and efficient risk identification method [26, 28]. Therefore, this paper measures risk based on the risk checklist and information entropy. On the basis of the systematical conclusion on the previous research achievements, this paper provides the following Risk Checklist 1 in Table 2.

**Table 2. Risk checklist 1.**

| Risk categories | No. | Risk items | ① | ② |
|---|---|---|---|---|
| Users | 1 | Whether the client senior organization supports the project development? | ● | ●\|○ |
| | 2 | Whether the use of the to-be-developed system will change the client workflow or organizational structure? | ● | ●\|○ |
| | 3 | Whether users are against the changes caused by the development of software project? | ● | ●\|○ |
| | 4 | Whether the use of the to-be-developed system will bring great convenience to users? | ● | ●\|○ |
| | 5 | How about the users' ability to pay the to-be-developed software system? | ● | ●\|○ |
| | 6 | Whether there are enough users to cooperate the project development? | ● | ●\|○ |
| | 7 | Whether users understand the technology of the to-be-developed system? | ● | ●\|○ |
| | 8 | Whether the development team understand the organization culture of users? | ● | ●\|○ |
| Requirement | 9 | Whether there are continuous requirement alterations? | ● | ●\|○ |
| | 10 | Whether there is a clear understanding on the system development requirements? | ● | ●\|○ |
| | 11 | Whether clients completely participate in the requirement research and analytical process? | ● | ●\|○ |
| | 12 | Whether the recorded requirements can be understood? | ● | ●\|○ |
| | 13 | Whether the developers and clients have the same understanding on the requirement specification? | ● | ●\|○ |
| | 14 | Whether there are technological requirements that are difficult to realize? | ● | ●\|○ |
| | 15 | Whether there are formal control process of requirement changes? | ● | ●\|○ |
| | 16 | Whether the budget changes with the requirement changes? | ● | ●\|○ |
| | 17 | Whether the time schedule changes with the requirement changes? | ● | ●\|○ |
| Project complexity and technology | 18 | Whether the project is relevant to the new technology or immature one? | ● | ●\|○ |
| | 19 | Whether the relevant technological complexity is high? | ● | ●\|○ |
| | 20 | Whether the project adopts the technology that has not been used by the project team before? | ● | ●\|○ |
| | 21 | How about the project scale? | ● | ●\|○ |
| | 22 | Whether the project team is experienced in this application field? | ● | ●\|○ |
| | 23 | Whether the program, method and tools support the development process? | ● | ●\|○ |
| Planning and control | 24 | Whether the project time schedule gears to actual circumstances? | ● | ●\|○ |
| | 25 | Whether the project milestone has been clearly defined? | ● | ●\|○ |
| | 26 | Whether the time schedule is stable? | ● | ●\|○ |
| | 27 | Whether the time schedule estimation method is based on historical data? | ● | ●\|○ |
| | 28 | Whether the time schedule estimation method is efficient when used in the past? | ● | ●\|○ |
| | 29 | Whether the budget is based on the actual estimation? | ● | ●\|○ |
| | 30 | Whether the budget is stable? | ● | ●\|○ |
| | 31 | Whether the budget estimation method is based on historical data? | ● | ●\|○ |
| | 32 | Whether the budget estimation method efficient when used in the past? | ● | ●\|○ |
| | 33 | Whether the project manager is experienced (according to the quantity and years of projects with similar scale) ? | ● | ●\|○ |
| | 34 | Whether the project is managed according to the schedule? | ● | ●\|○ |
| | 35 | Whether the project progress is under efficient monitoring (*e.g.* Regular status report, efficient project management technologies and tools *etc.*)? | ● | ●\|○ |
| | 36 | Whether the quality guarantee mechanism is defined? | ● | ●\|○ |

**Table 2. (Cont'd) Risk checklist 1.**

| | | | | |
|---|---|---|---|---|
| | 37 | Whether the time schedule hinders the quality? | ● | ●\|○ |
| Development team | 38 | How about the foundation level of the development team (eg. education level, professional background *etc.*)? | ● | ●\|○ |
| | 39 | How about the experience of the development team? | ● | ●\|○ |
| | 40 | Whether the development team understands the business of clients? | ● | ●\|○ |
| | 41 | Whether the staffs can obtain the relevant training for the skill which is need for the project? | ● | ●\|○ |
| | 42 | Whether the developers can participate in the development process from start to finish? | ● | ●\|○ |
| | 43 | Whether the turnover of the developers can ensure the continuity of the work? | ● | ●\|○ |
| | 44 | Whether the developers can focus on the development work of the project? | ● | ●\|○ |
| | 45 | Whether the project depends on several key staffs? | ● | ●\|○ |
| | 46 | Whether the staffs understand his and others roles? | ● | ●\|○ |
| | 47 | Whether the developers have right expectation on their work and feel comfortable? | ● | ●\|○ |
| | 48 | Whether staffs suitable to all the assigned work? | ● | ●\|○ |
| | 49 | Whether the staffs cooperate well through functional boundary? | ● | ●\|○ |
| | 50 | Whether the project members communicate well? | ● | ●\|○ |
| | 51 | Whether the project manager communicates well with the staffs? | ● | ●\|○ |
| | 52 | Whether the team members feel easy when asking for help from the manager? | ● | ●\|○ |
| organization environment of the development party | 53 | Whether the senior managers support the project? | ● | ●\|○ |
| | 54 | Whether the organization structure is stable? | ● | ●\|○ |
| | 55 | Whether there are adverse policies for the project? | ● | ●\|○ |
| | 56 | Whether there is internal conflict that influences the project? | ● | ●\|○ |

Note: ① means Before risk identification, ② means After risk identification, ● means the clause of this risk factor is unclear, namely the uncertainty is not eliminated, ○ means the clause of this risk factor is clear, namely the uncertainty has been eliminated.

### 4.3 Case Analysis

The software development risk measurement is a research topic with strong practice and its ultimate aim is to help the enterprise project management. This section carries out case analysis on the proposed measurement method. Company K undertakes the management information system of Y unit. This paper makes use of this case to analyze the application of this measurement method.

At first, Company K is not familiar with the detailed risk condition of this software project. Table 2 collects 56 items of risk factors. Known from the analysis in Section 4.1, nothing is known before the risk identification (shown as Table 3).

**Table 3. Check conditions of risk.**

| No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Risk status | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| No. | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 |
| Risk status | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| No. | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | |
| Risk status | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | |

According to Formula (4), $f_1 = f_2 = f_{56} = 1$ (Because the uncertainty of all the risk factors is not eliminated),

$$Risk_1 = 4.025. \tag{6}$$

To have an efficient management and control on the project, the company founds a risk management group to involve some staffs, client senior managers, client representatives, consultant experts and other persons in the contribution to the acquisition of risk data and the identification of the possible risk factors. After the risk identification, the conditions of some risk factors have been clear while some were still unclear. The details are shown as Table 4.

**Table 4. Risk status.**

| No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Inspection results | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ● | ● | ● | ○ | ○ | ○ | ○ | ● | ● | ○ | ○ |
| No. | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 |
| Inspection results | ○ | ○ | ○ | ● | ● | ○ | ● | ● | ● | ● | ● | ● | ● | ○ | ● | ● | ● | ● | ○ |
| No. | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | |
| Inspection results | ○ | ○ | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ○ | ○ | ● | ● | |

Namely,

$$f_i = \begin{cases} 0 & \text{where} \quad h \in s1 \quad \text{the uncertainty has been eliminated} \\ 1 & \text{where} \quad h \in s2 \quad \text{the uncertainty is not eliminated} \end{cases}$$

where $\begin{cases} s1 = 1,...,5,7,8,12,...,15,18,...,22,25,33,38,...,40,53,54 \\ s2 = 6,9,...,11,16,17,23,24,26,...,32,34,...,37,41,...,52,55,56 \end{cases}$.

According to Formula (3),

$$Risk_2 = 3.497. \tag{7}$$

After the above risk identification, there are still 33 unknown risk factors. The risk management team carries out the second risk identification and further investigates the risk conditions of the current project on the above basis. The obtained risk data are shown as Table 5.

**Table 5. Risk status.**

| No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Inspection results | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ● | ○ | ● | ○ | ○ | ○ | ○ | ● | ● | ○ | ○ |
| No. | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 |
| Inspection results | ○ | ○ | ○ | ● | ○ | ○ | ● | ○ | ● | ● | ○ | ○ | ● | ○ | ○ | ● | ○ | ● | ○ |
| No. | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | |
| Inspection results | ○ | ○ | ● | ○ | ○ | ● | ○ | ● | ○ | ● | ○ | ● | ● | ○ | ○ | ○ | ● | ○ | |

Namely,

$$f_i = \begin{cases} 0 & \text{where} \quad h \in s1 \quad \text{the uncertainty has been eliminated} \\ 1 & \text{where} \quad h \in s2 \quad \text{the uncertainty is not eliminated} \end{cases}$$

$$\text{where} \begin{cases} s1 = 1,...,5,7,8,10,12,...,15,18,...,22,24,25,27,30,31,33,34,36,38, \\ \quad ...,40,42,43,45,47,49,52,53,54,56 \\ s2 = 6,9,11,16,17,23,26,28,29,32,35,37,41,44,46,48,50,51,55 \end{cases}.$$

According to Formula (3),

$$Risk_3 = 2.944. \tag{8}$$

Seen from the results of three measurement results, $Risk_3 = 2.944 < Risk_2 = 3.497 < Risk_1 = 4.025$. It can be seen that the uncertainty before the risk information acquisition is maximum, so are the information entropy and risk. After obtaining the project risk factors status, we could get the risk information, which means the decrease of the uncertainty, so do the information entropy and risk. The larger the uncertainty is, the larger the information entropy and risk. Therefore, we can use information entropy to measure the risk of software development project and the entropy value to represents the risk.

## 5. AN IMPROVED MEASUREMENT MODEL

### 5.1 Improvements of the Model

Method in Section 4 measures risk in view of the information amount of software project risk factors. The acquisition of the information means the elimination of uncertainty, the decrease of risk and information entropy value, the entropy value can be used to describe the risk. This method is easy and feasible. However, it also owned its problems.

First, the elimination degree of uncertainties.

It is understandable to define the risk as maximum in case of unknowing the items of risk factors (namely●), because the uncertainty is maximum. After knowing the conditions of this item (namely○), the uncertainty of this item is eliminated. However, the risk after the complete elimination of the uncertainty obviously is lower than that of the partial elimination. Furthermore, this method fails to distinguish the complete elimination and partial elimination, so this kind of measurement is not accurate, for example, Risk Factor 18th in Table 2, "whether the project is relevant to the new technology or immature one?". "Being relevant to the new technology" and "being irrelevant to the new technology" both have definite answers while their risk are far from different. "Being irrelevant to the new technology" could be thought as no risk, namely the uncertainty is completely eliminated. While "being relevant to the new technology" obviously means owning risk, namely the uncertainty is only partially eliminated.

Therefore, on the one hand, the determination of one risk factor shall pay attention to whether there are certain answers (namely, whether information can be obtained); on

the other hand, it shall also distinguish the information with complete elimination of uncertainty from the information with partial elimination of uncertainty. After the adjustment of Table 2, Table 6 is as follows.

**Table 6. Risk checklist 2.**

| Risk categories | No. | Risk items | ① | Check conditions of risk | | |
|---|---|---|---|---|---|---|
| Users | 1 | Whether the client senior organization supports the project development? | ● | ○Average | ○support | ○Greatly support |
| | 2 | Whether the use of the to-be-developed system will change the client workflow or organizational structure? | ● | ○Greatly change | ○Little change | ○No |
| | 3 | Whether users are against the changes caused by the development of software project? | ● | ○Yes | ○Not obvious | ○No |
| | 4 | Whether the use of the to-be-developed system will bring great convenience to users? | ● | ○Average | ○Quite convenient | ○Very convenient |
| | 5 | How about the users' ability to pay the to-be-developed software system? | ● | ○Weak | ○Average | ○Strong |
| | 6 | Whether there are enough users to cooperate the project development? | ● | ○Less | ○Average | ○More |
| | 7 | Whether users understand the technology of the to-be-developed system? | ● | ○No | ○Average | ○Yes |
| | 8 | Whether the development team understand the organization culture of users? | ● | ○No | ○Average | ○Yes |
| Requirement | 9 | Whether there are continuous requirement alterations? | ● | ○Yes | ○Average | ○No |
| | 10 | Whether there is a clear understanding on the system development requirements? | ● | ○No | ○Average | ○Yes |
| | 11 | Whether clients completely participate in the requirement research and analytical process? | ● | ○No | ○Average | ○Yes |
| | 12 | Whether the recorded requirements can be understood? | ● | ○No | ○Average | ○Yes |
| | 13 | Whether the developers and clients have the same understanding on the requirement specification? | ● | ○No | ○most | ○Yes |
| | 14 | Whether there are technological requirements that are difficult to realize? | ● | ○Yes | ○Little | ○No |
| | 15 | Whether there is formal control process of requirement changes? | ● | ○No | ○Yes, but not perfect | ○Perfect |
| | 16 | Whether the budget changes with the requirement changes? | ● | ○No | ○Little | ○Yes |
| | 17 | Whether the time schedule changes with the requirement changes? | ● | ○No | ○Little | ○Yes |
| Project complexity and technology | 18 | Whether the project is relevant to the new technology or immature one? | ● | ○Yes | ○Little | ○No |
| | 19 | Whether the relevant technological complexity is high? | ● | ○High | ○Average | ○Low |
| | 20 | Whether the project adopts the technology that has not been used by the project team before? | ● | ○Yes | ○Yes, little | ○No |
| | 21 | How about the project scale? | ● | ○Large | ○Average | ○Small |
| | 22 | Whether the project team is experienced in this application field? | ● | ○No | ○Average | ○Yes |
| | 23 | Whether the program, method and tools support the development process? | ● | ○No | ○Average | ○Yes |
| Planning and control | 24 | Whether the project time schedule gears to actual circumstances? | ● | ○No | ○Average | ○Yes |
| | 25 | Whether the project milestone has been clearly defined? | ● | ○No | ○Yes, but not clearly | ○Yes |
| | 26 | Whether the time schedule is stable? | ● | ○No | ○Average | ○Yes |
| | 27 | Whether the time schedule estimation method is based on historical data? | ● | ○No | ○Basically | ○Yes |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 28 | Whether the time schedule estimation method is efficient when used in the past? | ● | ○Poor | ○Average | ○Great |
| | 29 | Whether the budget is based on the actual estimation? | ● | ○No | ○Average | ○Yes |
| | 30 | Whether the budget is stable? | ● | ○No | ○Average | ○Yes |
| | 31 | Whether the budget estimation method is based on historical data? | ● | ○No | ○Basically | ○Yes |
| | 32 | Whether the budget estimation method efficient when used in the past? | ● | ○Not ideal | ○Average | ○Yes |
| | 33 | Whether the project manager is experienced (according to the quantity and years of projects with similar scale)? | ● | ○No | ○Average | ○Yes |
| | 34 | Whether the project is managed according to the schedule? | ● | ○No | ○Basically | ○Yes |
| | 35 | Whether the project progress is under efficient monitoring (*e.g.* Regular status report, efficient project management technologies and tools *etc.*) ? | ● | ○No | ○Average | ○Yes |
| | 36 | Whether the quality guarantee mechanism is defined? | ● | ○No | ○Yes, but not perfect | ○Yes |
| | 37 | Whether the time schedule hinders the quality? | ● | ○Yes | ○Little | ○No |
| Development team | 38 | How about the foundation level of the development team (*e.g.* education level, professional background *etc.*)? | ● | ○Weak | ○Average | ○Strong |
| | 39 | How about the experience of the development team? | ● | ○Poor | ○Average | ○Great |
| | 40 | Whether the development team understands the business of clients? | ● | ○No | ○Basically | ○Yes |
| | 41 | Whether the staffs can obtain the relevant training for the skill which is need for the project? | ● | ○No | ○Average | ○Yes |
| | 42 | Whether the developers can participate in the development process from start to finish? | ● | ○No | ○Average | ○Yes |
| | 43 | Whether the turnover of the developers can ensure the continuity of the work? | ● | ○No | ○Average | ○Yes |
| | 44 | Whether the developers can focus on the development work of the project? | ● | ○No | ○Average | ○Yes |
| | 45 | Whether the project depends on several key staffs? | ● | ○Yes | ○Partial | ○No |
| | 46 | Whether the staffs understand his and others roles? | ● | ○No | ○Average | ○Yes |
| | 47 | Whether the developers have right expectation on their work and feel comfortable? | ● | ○No | ○Average | ○Yes |
| | 48 | Whether staffs suitable to all the assigned work? | ● | ○No | ○Average | ○Yes |
| | 49 | Whether the staffs cooperate well through functional boundary? | ● | ○No | ○Average | ○Yes |
| | 50 | Whether the project members communicate well? | ● | ○No | ○Average | ○Yes |
| | 51 | Whether the project manager communicates well with the staffs? | ● | ○No | ○Average | ○Yes |
| | 52 | Whether the team members feel easy when asking for help from the manager? | ● | ○No | ○Average | ○Yes |
| organization environment of the development party | 53 | Whether the senior managers support the project? | ● | ○No | ○Little | ○Yes |
| | 54 | Whether the organization structure is stable? | ● | ○No | ○Partial | ○Yes |
| | 55 | Whether there are adverse policies for the project? | ● | ○Yes, and the influence is serious | ○Average | ○No |
| | 56 | Whether there is internal conflict that influences the project? | ● | ○Yes, and the influence is serious | ○Average | ○No |

Note: ① represents before risk identification, or represents a circumstance that the risk remains unknown after risk inspection, which means no eliminated uncertainty.

Table 2 fails to refine the elimination degrees of risk after the risk identification while Table 6 classifies the elimination degrees of risk after the risk identification into 3

levels, columns 5-7. Column 7 means complete elimination uncertainty and the value of this risk item is 0; columns 6 and 5 mean partial elimination uncertainty and the risk still exists. However, the eliminated uncertainty in column 6 is larger than that in column 5. Namely, column 6 means more eliminated uncertainty and column 5 means little eliminated uncertainty. In this way, risk of column 6 is lower than that of column 5.

Second, the importance level of risk factors.

In Section 4, each risk factor is considered as equally important, but it is different in reality because some factors are very important and some are less important. This research determines the importance level of risk factors by risk prioritizing, as shown in Section 5.2.

After the inclusion of the elimination degree of uncertainties and the importance level of risk factors, Formula (3) is modified as below,

**Risk measurement model 2,**

$$Risk = H_j(f_1, f_2, ..., f_n) = -\sum_{j=1}^{n}(\rho_j \ln \rho_j) \tag{9}$$

where $\rho_j = \dfrac{f_j}{\sum_{k=1}^{n} f_k}$, $f_j = \dfrac{1}{m} \times \sum_{i=1}^{m}((1 - EDoU_{ij}) \times IDoRF_j)$, $j = 1, 2, ..., n$.

Namely,

$$Risk = -\sum_{j=1}^{n}\left(\frac{\sum_{i=1}^{m}((1 - EDoU_{ij}) \times IDoRF_j)}{\sum_{k=1}^{n}(\sum_{l=1}^{m}((1 - EDoU_{lk}) \times IDoRF_k))} \ln\left(\frac{\sum_{i=1}^{m}((1 - EDoU_{ij}) \times IDoRF_j)}{\sum_{k=1}^{n}(\sum_{l=1}^{m}((1 - EDoU_{lk}) \times IDoRF_k))}\right)\right) \tag{10}$$

where $EDoU_{ij}$ represents the elimination degree of uncertainties of the $j$th risk item determined by the $i$th risk inspector, $IDoRF_j$ represents the important degree of risk factor in the $j$th risk items.

Formula (10) shows the risk value of a software development project.

### 5.2 Case Analysis

The information technology research institution of Y University (hereinafter noted as A) mainly is occupied in the software development and has many successful development precedents in university informatization, hospital informatization management and other fields. This institution undertook the E-Government software development project of one country (hereinafter noted as B) in 2011. This paper took this case to illustrate the application of the improved measurement method in this section.

A appointed C as the project manager to launch the project requirement research, the project plan, risk analysis and a series of work. C called together 3 members from the project group, 3 representatives of Organization B, 2 industrial consultants, and 1 expert in this field to set up the 10-member risk management team. According to the Risk Checklist (shown as Table 6), the risk information was obtained as shown in Fig. 1.

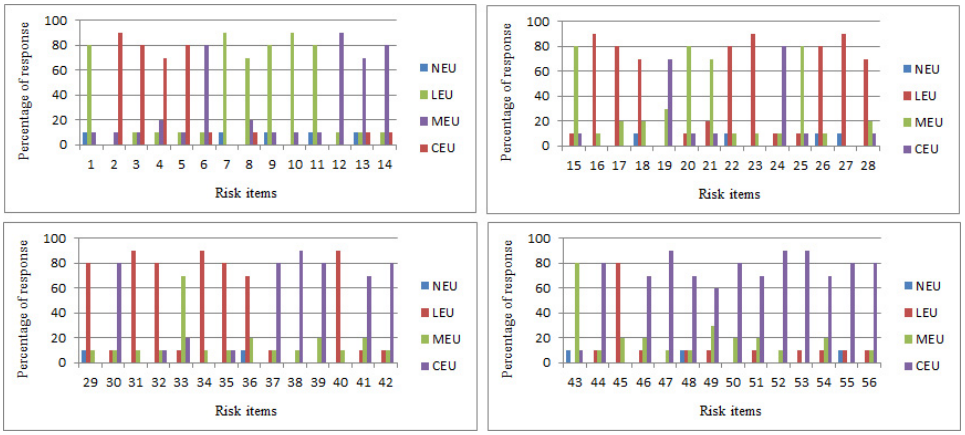



Fig. 1. Response for eliminated degree of uncertainty.

NEU, LEU, MEU and CEU mean respectively no eliminated uncertainty, little eliminated uncertainty, more eliminated uncertainty and complete eliminated uncertainty. Values of the elimination degree are as follows,

$$EDoU_i = \begin{cases} 0 & \text{no eliminated uncertainty} \\ 0.35 & \text{little eliminated uncertainty} \\ 0.7 & \text{more eliminated uncertainty} \\ 1 & \text{complete eliminated uncertainty} \end{cases}.$$

The values of $EDoU_{ij}$ in Formula (10) are obtained in the above. Values of $IDoRF_j$ in Formula (10) can be obtained by risk prioritizing. The risk prioritizing has been studied by many researchers. The founder of software project risk management, Boehm [5], investigated some large-scale projects and summed up a list of top ten risks in software development project (as shown in Table 7). The internationally renowned expert of software project risk management, Schmidt [3], carried out an international Delphi investigation for identifying and prioritizing software project risk factors. He invited 45 experienced experts from 3 countries and regions (Hong Kong, Finland and the United States) in this research, and divided these experts into three independent groups (HKG, FIN and USA) according to their nationalities. HKG, FIN and USA Group had 11, 13 and 21 experts respectively. The research encompassed three stages. Firstly, each expert proposed at least 6 items of risk factors in the brainstorming. Next, all proposed risk factors were summarized and filtered to identify the final list of risks. Finally, risk factors on the list were prioritized according to the importance level. HKG, FIN and USA Group identified 15, 17 and 23 items of risk factors respectively, and the items and orders are shown in Table 8. Addison and Vallabh [4], a researcher of the world famous KPMG further summarized 14 items of risk factors based on previous findings in this field. They also invited 70 software project managers in their research. Managers were asked to identify the importance level of each risk factor from 5 levels (completely unimportant, not very important, important, very important and extremely important). After obtaining the feedback data, two researchers sorted out the ten most important risk factors (as shown in Table 9).

Obviously, Tables 7-9 show differences in the number of risk items, contents of risk factors and risk prioritizing, showing that views of risks differed among different scholars and practitioners. These views also share similarities. For example, in Table 8, outcomes of HKG, FIN, and USA Group had 11 identical risk factors. Same risk prioritizing is also observed, for example, Item 1 and Item 12 are the same in HKG and USA Group,

**Table 7. A top ten list of software risk items [5].**

| Ranking | Risk item |
|---|---|
| 1 | Personnel shortfalls |
| 2 | Unrealistic schedules and budgets |
| 3 | Developing the wrong software functions |
| 4 | Developing the wrong user interface |
| 5 | Gold plating |
| 6 | Continuing stream of requirements changes |
| 7 | Shortfalls in externally furnished components |
| 8 | Shortfalls in externally performed tasks |
| 9 | Real-time performance shortfalls |
| 10 | Strainning computer science capabilities |

**Table 8. Rankings of risk items [3].**

| Ranks | | | | Risk items |
|---|---|---|---|---|
| Composite | HKG | USA | FIN | |
| 1 | 1 | 1 | 2 | Lack of top management commitment to the project |
| 2 | 3 | 4 | 8 | Failure to gain user commitment |
| 3 | 7 | 2 | 6 | Misunderstanding the requirement |
| 4 | 2 | 6 | 11 | Lack of adequate user involvement |
| 5 | 13 | 11 | 3 | Lack of required knowledge/skills in the project personnel |
| 6 | 8 | 14 | 9 | Lack of frozen requirements |
| 7 | 5 | 10 | 19 | Changing scope/objectives |
| 8 | 12 | 12 | 13 | Introduction of new technology |
| 9 | 9 | 7 | 23 | Failure to manage end user expections |
| 10 | 15 | 13 | 15 | Insufficient/inappropriate staffing |
| 11 | 10 | 16 | 22 | Conflict between user departments |
| | 4 | | | Lack of cooperation from users |
| | 5 | | | Change in ownership or senior management |
| | 11 | | 17 | Staffing volatility |
| | 14 | | | Lack of effective development process/methodology |
| | | 3 | 4 | Not managing change properly |
| | | 5 | 1 | Lack of effective project management skills |
| | | 8 | | Lack of effective project management methodology |
| | | 9 | | Unclear/misunderstood scope/objectives |
| | | 15 | 20 | Improper definition of roles and responsibilities |
| | | 17 | | Number of organizational units involved |
| | | | 5 | No planning or inadequate planning |
| | | | 7 | Artificial deadlines |
| | | | 12 | Multi-vendor projects complicate dependcies |
| | | | 10 | Lack of "people skills" in project leadership |
| | | | 14 | Trying new development method/technology during important project |
| | | | 18 | Bad estimation |
| | | | 16 | New and/or unfamiliar subject matter for both users and developers |
| | | | 21 | Poor or nonexistent control |

**Table 9. Importance of risks [4].**

| Ranking | Risk factors |
|---|---|
| 1 | Unclear or misunderstood scope/objectives |
| 2 | Misunderstanding the requirements |
| 3 | Failure to gain user involvement |
| 4 | Lack of senior management commitment |
| 5 | Developing the wrong software functions |
| 6 | Unrealistic schedules and budgets |
| 7 | Continuous requirement changes |
| 8 | Inadequate knowledge/skills |
| 9 | Lack of effective project management methodology |
| 10 | Gold plating |

and Item 15 is the same in HKG and FIN Group. This is also observed in Tables 7-9. For example, Item 2 in Table 7 and Item 6 in Table 9 are the same. Item 6 in Table 7 and Item 7 in Table 9 are the same. In Table 8, Item 5 in USA Group is the same with Item 1 in FIN Group as well as Item 9 in Table 9. This indicates that the investigation results of 5 groups in Tables 7-9 share some same findings in terms of contents of risk factors as well as risk prioritizing. Thus, different scholars and practitioners have both differences and commonalities in viewing risks of software project.

In traditional research of risk measurement and assessment, the importance level of risk factors was often set by a handful of experts based on subjective judgment(such as 5 experts in Literature [33] and 3 in Literature [34]). This led to significant subjective interference. In order to minimize the subjective influence of human factors, this research applies a comprehensive integrated approach to combine the findings of the Risk Checklist Table 6 and Tables 7-9 with the principle of same or similar contents of risk factors (as shown in Table 10).

However, these tables show certain mismatching in terms of the number of risk item or contents of risk factors. For example, Item 2 in Table 6 cannot be matched in Tables 7 and 9, showing that this factor was considered as risky by the author but not risk by Boehm and Addison, or not in the top ten risks. In Table 8, the three investigation groups show 18 mismatching risk factors. According to the table, it is found that Schmidt does not prioritize these items. To address this, this research uses the method of interpolation to prioritize all risks. For risk factors found in Table 6 but not in Tables 7-9, a value is inserted. As these risk factors do not exist in Tables 7-9, it can be concluded that these items are not taken as risky by these three literatures, or have an importance level lower than all items in Tables 7-9. Thus, the number of inserted values should be larger than the total number of risk factors in Tables 7-9. In Tables 7 and 9, there are 10 risk factors. HKG, FIN and USA Group identified 15, 17 and 23 items of risk factors respectively in Table 8. Hence, the numerical value 25 is inserted in Table 10, which solves the prioritizing of all risk factors in Table 6 (as shown in Column 3 of Table 10).

After the solution of importance level prioritizing for risk factors in Table 6, in order to calculate the risk values of a project, the importance level should be quantified to numerical values. This research quantifies the importance level according to the ranks 1, 2, 3, 4…, to numerical value 1, 0.99. 0.98, 0.97… (as shown in Column 2 of Table 10), thus obtaining the value of $IDoRF_j$ in Formula (10).

**Table 10. My rankings of risk items and the value.**

| No. | Value | Ranks | | | | | |
|---|---|---|---|---|---|---|---|
| | | Composite | Boehm | HKG | USA | FIN | Addison |
| 1 | 0.95 | 6 | 25 | 3 | 4 | 8 | 4 |
| 2 | 0.7 | 31 | 25 | 10 | 16 | 22 | 25 |
| 3 | 0.7 | 31 | 25 | 10 | 16 | 22 | 25 |
| 4 | 0.66 | 35 | 25 | 4 | 25 | 25 | 25 |
| 5 | 0.54 | 47 | 25 | 25 | 25 | 25 | 25 |
| 6 | 0.94 | 7 | 25 | 2 | 6 | 11 | 3 |
| 7 | 0.54 | 47 | 25 | 25 | 25 | 25 | 25 |
| 8 | 0.54 | 47 | 25 | 25 | 25 | 25 | 25 |
| 9 | 0.94 | 7 | 6 | 5 | 10 | 19 | 7 |
| 10 | 1 | 1 | 3 | 7 | 2 | 6 | 2 |
| 11 | 0.99 | 2 | 3 | 2 | 6 | 11 | 3 |
| 12 | 0.97 | 4 | 3 | 7 | 2 | 6 | 25 |
| 13 | 0.97 | 4 | 3 | 7 | 2 | 6 | 25 |
| 14 | 0.78 | 23 | 25 | 12 | 12 | 13 | 25 |
| 15 | 0.9 | 11 | 6 | 9 | 7 | 23 | 25 |
| 16 | 0.83 | 18 | 25 | 25 | 3 | 4 | 25 |
| 17 | 0.83 | 18 | 25 | 25 | 3 | 4 | 25 |
| 18 | 0.9 | 11 | 25 | 12 | 12 | 13 | 8 |
| 19 | 0.9 | 11 | 25 | 12 | 12 | 13 | 8 |
| 20 | 0.63 | 38 | 25 | 25 | 25 | 25 | 8 |
| 21 | 0.54 | 47 | 25 | 25 | 25 | 25 | 25 |
| 22 | 0.57 | 44 | 25 | 25 | 25 | 16 | 25 |
| 23 | 0.59 | 42 | 25 | 14 | 25 | 25 | 25 |
| 24 | 0.92 | 9 | 2 | 25 | 25 | 7 | 6 |
| 25 | 0.77 | 24 | 25 | 25 | 8 | 25 | 9 |
| 26 | 0.83 | 18 | 25 | 25 | 3 | 4 | 25 |
| 27 | 0.73 | 28 | 25 | 14 | 25 | 25 | 6 |
| 28 | 0.73 | 28 | 25 | 14 | 25 | 25 | 6 |
| 29 | 0.79 | 22 | 2 | 25 | 25 | 25 | 6 |
| 30 | 0.83 | 18 | 25 | 25 | 3 | 4 | 25 |
| 31 | 0.73 | 28 | 25 | 14 | 25 | 25 | 6 |
| 32 | 0.59 | 42 | 25 | 14 | 25 | 25 | 25 |
| 33 | 0.92 | 9 | 25 | 25 | 5 | 1 | 9 |
| 34 | 0.55 | 46 | 25 | 25 | 25 | 21 | 25 |
| 35 | 0.7 | 31 | 25 | 14 | 25 | 25 | 9 |
| 36 | 0.7 | 31 | 25 | 14 | 25 | 25 | 9 |
| 37 | 0.64 | 37 | 25 | 25 | 25 | 7 | 25 |
| 38 | 0.76 | 25 | 25 | 15 | 13 | 15 | 25 |
| 39 | 0.76 | 25 | 25 | 15 | 13 | 15 | 25 |
| 40 | 0.57 | 44 | 25 | 25 | 25 | 16 | 25 |
| 41 | 0.54 | 47 | 25 | 25 | 25 | 25 | 25 |
| 42 | 0.87 | 14 | 1 | 11 | 25 | 17 | 25 |
| 43 | 0.87 | 14 | 1 | 11 | 25 | 17 | 25 |
| 44 | 0.87 | 14 | 1 | 11 | 25 | 17 | 25 |
| 45 | 0.87 | 14 | 1 | 11 | 25 | 17 | 25 |
| 46 | 0.62 | 39 | 25 | 25 | 15 | 20 | 25 |
| 47 | 0.62 | 39 | 25 | 25 | 15 | 20 | 25 |
| 48 | 0.76 | 25 | 25 | 15 | 13 | 15 | 25 |
| 49 | 0.62 | 39 | 25 | 25 | 15 | 20 | 25 |
| 50 | 0.65 | 36 | 25 | 25 | 25 | 25 | 5 |

According to Formula (10),

$$Risk_3 = 3.721. \tag{11}$$

Through the risk measurement in the previous stage, C found that the project risk was huge, mainly manifesting in the insufficient support of the senior organization of B Unit to the project (which led to the insufficient participation of users in the requirement research and analysis), the vague understanding of the development system requirement, the causal requirement and changes proposed by the leaders (which resulted in the frequent changes of the requirement), failing to comprehend the development techniques and so on. C reorganized the risk management team to carry out a further research and strengthen the communication, obtaining the following risk information (as shown in Fig. 2).

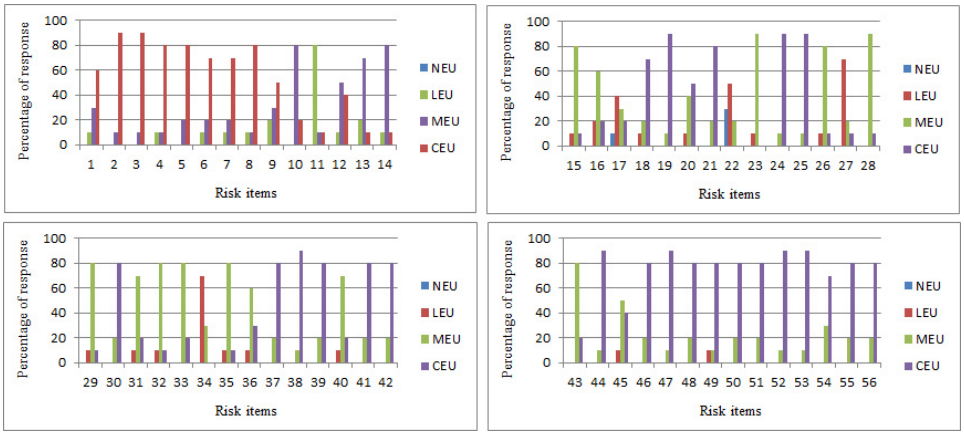



Fig. 2. Response for eliminated degree of uncertainty.

According to Formula (10),

$$Risk_4 = 3.644. \tag{12}$$

Measurement data demonstrates that efficient risk measurement and management can decrease the project risk. After obtaining the project risk factors status, we could get the risk information, which means the decrease of the uncertainty, information entropy and risk. The more the information amount of the risk factors are, the more the eliminated uncertainty is, the lower the information entropy is and the smaller the risk is. The above case analysis indicates that the model proposed in Formula (10) is feasible and can effectively measure the software development project risk.

## 6. DISCUSSION AND ADVANTAGES COMPARED WITH OTHER METHODS

In view of the obtained information of risk factors, information entropy is used to measure the software development project risk in this paper. The paper proposes a mea-

surement model in Formula (10) and its reasonability has been discussed in Section 3 to 5. To sum up, the essential feature of risks is the uncertainty. Information entropy is an effective measurement method of uncertainty. Therefore, the fundamental logic of the model is reasonable.

In traditional research of risk measurement and assessment, the probability of risk occurrence and its influence is subjectively set by experts, leading to significant influence of human factors. This tends to cause less objective and effective measurement results. Methods shown in Table 11 all have this defect, such as 5 experts in Literature [33] and 3 in Literature [34].

The model proposed in our research (shown as Formula (10)) only consists of two parameters, $EDoU_{ij}$ and $IDoRF_j$. No parameter is graded and determined by experts. The first parameter is obtained by the actual survey of the project risk management group according to the Risk Checklist in Table 6. The second parameter is obtained by the integration of this research finding and the findings of Boehm, Schmidt, and Addison. Although the findings of Boehm, Schmidt, and Addison (as shown in Tables 7-9) involve subjective judgment, these findings are reached by the actual survey and investigation of hundreds of experienced project managers and hundreds of software development projects all over the globe, thus showing certain reliability. Thus, the negative impact from subjective factors in findings of Boehm, Schmidt, and Addison is much less significant than that of the method in Table 11, which only invited a handful of experts to grade and identify the probability of risk occurrence and its influence. Therefore, the risk measurement method proposed in this research (as shown in Formula (10)) show more advantages than the traditional methods (as shown in Table 11), and this method can minimize the influence of subjective factors. The case analysis proves that the risk measurement method proposed in this paper has sound operability and can provide objective data to the software project managers' decision-making.

**Table 11. Several typical methods.**

| No. | Methods | Representative literatures | Shortcomings |
|---|---|---|---|
| 1 | Dale Karolak | [13] | Risk occurrence probability and impact of subjective assessment |
| 2 | Bayesian Network | [35] | Risk occurrence probability and losses are assessed by experts |
| 3 | Grey Clustering Method | [33] | Risk is scored by risk experts |
| 4 | BP neural network | [36] | Major parameters input manually |
| 5 | interval elementary dependent function | [37] | Risk is scored by risk experts |
| 6 | triangular fuzzy number multi-attribute decision making | [34] | The same as above |
| 7 | least squares support vector machine | [38] | The same as above |

## 7. CONCLUSIONS

The traditional risk measurement method usually assesses in terms of the occurrence probability and the loss of the risk factors, which usually are assessed by experts subjectively. Therefore, they are greatly influenced by artificial factors. In this way, it is difficult to realize the objective and effective measurement. This paper proposes one risk

measurement method of software development project based on information entropy. This method makes use of information entropy to measure information amount of risk factor. The paper concludes that the larger the information amount of the risk factors is, the larger the removed uncertainty is, the smaller the information entropy is and the smaller the risk is. In short, there is a positive correlation between risk and entropy. Compared with other risk measurement methods, this method manifests certain advantage, which makes up for the disadvantages of the subjective assessment in terms of occurrence probability and impact degree in previous studies. The case analysis prove the rationality and feasibility of this method.

## ACKNOWLEDGEMENTS

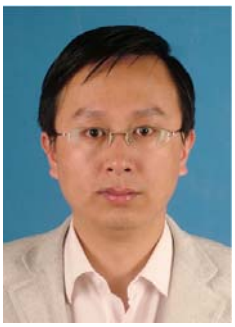
This work was supported by National Natural Science Foundation of China (Grant No. 61263022, 61303234 and 71362016), China Postdoctoral Science Foundation (Grant No. 2012M521722), National Social Science Foundation of China (Grant No. 12XTQ-012), Humanities and Social Sciences Youthful Foundation of the Ministry of Education of China (Grant No. 11YJCZH073), Natural Science Foundation of Yunnan Province of China (Grant No. 2010ZC100), Philosophy and Social Sciences of Planning Project of Yunnan Province (Grant No. QN201210), and Introduction of Talents Project of Science Research Foundation of Yunnan University of Finance and Economics (Grant No. YC-2012D07). The author would like to thank the anonymous reviewers and the editors for their suggestions.

## REFERENCES

1. T. W. Kwan and H. K. N. Leung, "A risk management methodology for project risk dependencies," *IEEE Transactions on Software Engineering*, Vol. 37, 2011, pp. 635-648.
2. W.-M. Han, "Validating differential relationships between risk categories and project performance as perceived by managers," *Empirical Software Engineering*, Vol. 18, 2013, pp. 1-11.
3. R. Schmidt, K. Lyytinen, M. Keil, and P. Cule, "Identifying software project risks: an international delphi study," *Journal of Management Information Systems*, Vol. 17, 2001, pp. 5-36.
4. T. Addoison and S. Vallabh, "Controlling software project risks – an empirical study of methods used by experienced project managers," in *Proceedings of Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on Enablement through Technology*, 2002, pp. 128-140.
5. R. R. B. Boehm, "Theory-W software project management: Principles and examples," *IEEE Transactions on Software Engineering*, Vol. 15, 1989, pp. 902-916.
6. B. W. Boehm, "Software risk management: principles and practices," *IEEE Software*, Vol. 8, 1991, pp. 32-41.
7. Q. Zheng, X. Wenhua, H. Yi, and T. Jing, *Software Project Management*, 2nd ed., Tsinghua University Press, Beijing, 2009.
8. L. Wallace, A. Rai, and M. Keil, "How software project risk affects project perfor-

mance: an investigation of the dimensions of risk and an exploratory model," *Decision Sciences*, Vol. 35, 2004, pp. 289-321.
9. J. Ropponen and K. Lyytinen, "Components of software development risk: how to address them? A project manager survey," *IEEE Transactions on Software Engineering*, Vol. 26, 2000, pp. 98-112.
10. J. Ropponen and K. Lyytinen, "Can software risk management improve system development: an exploratory study," *European Journal of Information Systems*, Vol. 6, 1997, pp. 41-50.
11. M. Keil, P. E. Cule, K. Lyytinen, and R. C. Schmidt, "A framework for identifying software project risks," *Communications of the ACM*, Vol. 41, 1998, pp. 76-83.
12. Boehm, *Software Risk Management*, 1989, Piscataway, IEEE Computer Society Press.
13. D. W. Karolak, *Software Engineering Risk Management: A Just-in-Time Approach*, 1998, IEEE, USA.
14. Kontio, *The Riskit Method for Software Risk Management*, in Computer Science Technical Reports, 1996, College Park, Maryland.
15. L. Salmeron, "Forecasting risk impact on ERP maintenance with augmented fuzzy cognitive maps," *IEEE Transactions on Software Engineering*, Vol. 38, 2012, pp. 439-452.
16. J. P. Li, M. L. Li, D. S. Wu, and H. Song, "An integrated risk measurement and optimization model for trustworthy software process management," *Information Sciences*, Vol. 191, 2012, pp. 47-60.
17. A. Herrmann and B. Paech, "Practical challenges of requirements prioritization based on risk estimation," *Empirical Software Engineering*, Vol. 14, 2009, pp. 644-684.
18. R. Shatnawi, "A quantitative investigation of the acceptable risk levels of object-oriented metrics in open-source systems," *IEEE Transactions on Software Engineering*, Vol. 36, 2010, pp. 216-225.
19. A. Sharma and A. Gupta, "Impact of organisational climate and demographics on project specific risks in context to Indian software industry," *International Journal of Project Management*, Vol. 30, 2012, pp. 176-187.
20. Y. Fu, M. Li, and F. Chen, "Impact propagation and risk assessment of requirement changes for software development projects based on design structure matrix," *International Journal of Project Management*, Vol. 30, 2012, pp. 363-373.
21. X. Shi, H. Tsuji, and S. Zhang, "Eliciting experts' perceived risk of software offshore outsourcing incorporating individual heterogeneity," *Expert Systems with Applications*, Vol. 38, 2011, pp. 2283-2291.
22. A. Appari and M. Benaroch, "Monetary pricing of software development risks: A method and empirical illustration," *Journal of Systems and Software*, Vol. 83, 2010, pp. 2098-2107.
23. G. Büyüközkan and D. Ruan, "Choquet integral based aggregation approach to software development risk assessment," *Information Science*, Vol. 180, 2010, pp. 441-451.
24. A. Benlian and T. Hess, "Opportunities and risks of software-as-a-service: Findings from a survey of IT executives," *Decision Support Systems*, Vol. 52, 2011, pp. 232-246.
25. S. J. Huang, C. Y. Lin, and N. H. Chiu, "Fuzzy decision tree approach for embed-

ding risk assessment information into software cost estimation model," *Journal of Information Science and Engineering*, Vol. 22, 2006, pp. 297-313.
26. M. Keil, L. Li, L. Mathiassen, and G. Zheng, "The influence of checklists and roles on software practitioner risk perception and decision-making," *The Journal of Systems and Software*, Vol. 81, 2008, pp. 908-919.
27. H. R. Costa, M. D. Barros, and G. H. Travassos, "Evaluating software project portfolio risks," *The Journal of Systems and Software*, Vol. 80, 2007, pp. 16-31.
28. P. L. Bannerman, "Risk and risk management in software projects: A reassessment," *The Journal of Systems and Software*, Vol. 81, 2008, pp. 2118-2133.
29. C. E. Shannon, "The mathematical theory of communication," *The Bell System Technical Journal*, Vol. 27, 1948, pp. 379-423, 623-656.
30. W.-M. Han and S.-J. Huang, "An empirical analysis of risk components and performance on software projects," *Journal of Systems and Software*, Vol. 80, 2007, pp. 42-50.
31. H. Barki, S. Rivard, and J. Talbot, "Toward an assessment of software development risk," *Journal of Management Information Systems*, Vol. 10, 1993, pp. 203-225.
32. F. J. Heemstra and R. J. Kusters, "Dealing with risks: a practical approach," *Journal of Information Technology*, Vol. 11, 1996, pp. 333-346.
33. F. Z. Ge, T. Peng, and G. F. Guo, "Application of grey clustering method in software risk assessment," *Computer Engineering and Applications*, Vol. 45, 2009, pp. 78-90.
34. L. Yang, N. Li, and Y. Y. He, "Application of triangular fuzzy number multiattribute decision making method in software project risk evaluation," *Computer Engineering and Applications*, Vol. 46, 2010, pp. 246-248.
35. A. G. Tang, R. L. Wang, and C. H. Hu, "Application of Bayesian networks in software project risk assessment," *Computer Engineering and Applications*, Vol. 46, 2010, pp. 62-65.
36. C. Zhao, Q. Zeng, Y. Yang, and J. Yang, "Research on risk assessment of software project based on BP neural network," *Application Research of Computers*, Vol. 26, 2009, pp. 3767-3770.
37. L. Yang and N. Li, "Application of interval elementary dependent function in software project risk evaluation," *Computer Engineering and Applications*, Vol. 45, 2009, pp. 196-198.
38. W. Wang, G. J. Zhao, and Q. Li, "Novel intelligent method of software project risk assessment," *Computer Engineering and Applications*, Vol. 43, 2007, pp. 8-11.



**Rong Jiang (姜茸)** is an Associate Professor at the School of Information, Yunnan University of Finance and Economics, China. He is also serving as an Academic Leader at School of Software, Yunnan University. He received his Ph.D. in System Analysis and Integration from the School of Software at Yunnan University. He has published more than 20 papers and four text books, and has gotten more than 30 prizes in recent six years. His current projects are in the areas of cloud computing, software engineering, software project management, risk measurement, information entropy, and systems science, *etc.*