

Software Project Risk and Management System Function Analysis

Yan Changshun, Shao Yong*

Faculty of Information Technology, Beijing University of Technology, Beijing, China

Email address:

yuewuxing@bjut.edu.cn (Yan Changshun), shaoyong@bjut.edu.cn (Shao Yong)

*Corresponding author

To cite this article:

Yan Changshun, Shao Yong. Software Project Risk and Management System Function Analysis. *American Journal of Information Science and Technology*. Vol. 6, No. 3, 2022, pp. 70-74. doi: 10.11648/j.ajist.20220603.14

Received: September 15, 2022; **Accepted:** October 18, 2022; **Published:** October 20, 2022

Abstract: Nowadays, with the rapid development of science and technology and the trend of economic globalization, more and more companies are trying to establish online industrial chains, clients in different servers or official websites of companies. In the field of software development, software risk involves all stages of the software development process, and all software projects have risks. In the whole software development process, every development link should incorporate the idea of risk management. It is an important content of risk management to transfer risk-based thinking to the participants in every link of software development, which requires the active participation of managers in every link. In order to enable managers in every link to participate in risk management activities, a deep understanding of software project risks can effectively help development teams or software project managers avoid possible risks, reduce losses caused by risks or monitor risks that have not yet occurred. In this paper, the research status, influencing factors, management steps and common methods of software project risk management at home and abroad are introduced in detail, and the functional suggestions of the risk management system are given. It is hoped that the work of this paper can give corresponding help to software project risk managers.

Keywords: Software Project, Risk Management, Step and Method

1. Introduction

After consulting a number of documents, we know that although the software development technology is constantly improving and the software project management method is more and more perfect, the failure rate of its projects has remained basically unchanged at about 40% - 50% in the past two decades [1, 2]. If a software development team rashly develops without considering the risks, it is easy to cause the failure of the whole development project due to minor reasons such as development progress, capital budget, technology maturity, quality of software achievements, and even communication errors [3, 4]. Therefore, it is not difficult to see that risk management is a key activity in the software development process. Risk management from the early stage of the project can predict the harm caused by risk as soon as possible, so as to avoid or reduce losses [5].

Because of the particularity of software project itself, software management cannot be carried out completely

according to the traditional project. The research on risk management of software projects began in the late 1960s. Because the IT industry has just started, the research level is not deep. It was not until the late 1980s that software project risk management emerged as a discipline. Subsequently, with the development of the software industry itself, the theory of project risk management in the field of software development is gradually improving.

At present, there are several classic risk management theoretical systems:

In 1989, Boehm divided risk management into two stages in his book software risk management. The first stage was the risk assessment stage, including three steps: risk identification, risk analysis and risk prioritization. The latter stage is the risk control stage, which includes three steps: formulating the risk management plan, risk solving and risk monitoring [6, 7].

The risk management system of the Software Engineering Institute of Carnegie Mellon University consists of five steps: risk identification, risk analysis, risk planning, risk tracking and risk control. It advocates continuous risk management,

that is, the above five steps should be continuously circulated throughout the whole software life cycle. SEI researched and developed Capability Maturity Model Integration on the basis of CMM in 2002. CMMI divides the software development capability of software enterprises into five levels, which are mainly used to guide the improvement of software development process and evaluate software development capability. Software risk management is included in the third level, which integrates software risk management into the process of software project management. This system is improved on the basis of Boehm and reduces the difficulty of calculation. However, CMMI evaluation needs a lot of money, takes a long time and is complex, so it is not suitable for small and medium-sized enterprises [8, 9].

In 1998 Hall proposed a six discipline risk management model. The model divides software risk management into six disciplines, epwmid. Where: e refers to the long-term plan for researching software products; P is a plan, which allocates resources to software projects; W stands for work, which refers to the implementation of product plan; M represents measurement, and adjusts the project plan by comparing the difference between the expected value and the actual value; I stands for improvement, learning from past software projects, and finding solutions by analyzing the gaps between standards and the project itself; D is representative discovery, based on Uncertainty Evaluation and thinking about confusion, considering the balance of opportunities and risks, guiding the change of project plan [9-11].

At present, the theoretical research level of software risk management in China is no less than that in developed countries, but the domestic research focuses on mathematical calculation and fails to form a systematic and complete theoretical system of software risk management. And because of its late start and lack of practical application, some small and medium-sized enterprises do not have the ability of software risk management, or it is difficult to carry out systematic and scientific risk management [12-14].

2. Risk Factors of Software Project

The core of Boehm's software risk management theory is to maintain and update the top ten risk lists. Boehm summarized the top ten risks of software projects based on the analysis and research of large software projects, which are still very common risks in the risk management of software projects today, and can be used as a reference for project risk management [15]. Boehm's top ten risks list includes:

1. Personnel shortage;
2. Unrealistic schedule planning and budget;
3. Wrong software functions are developed;
4. Developed wrong user interface;
5. Add unnecessary non functional requirements;
6. Frequent changes in requirements;
7. Insufficient outsourcing parts;
8. Gaps caused by externally executed tasks;
9. Insufficient real-time performance;
10. Computer power is limited.

Different software risk events have different probabilities at different stages of software development. In the stage of requirement analysis, the high risk is that customer requirements change frequently, which makes it difficult to determine software functional requirements; The customer's demand does not match the project duration or budget; Unclear customer needs; It is difficult to clearly describe the customer's needs in the requirements document. In the system design stage, there are risks: unreasonable task allocation, demanding construction period or quality from customers or project leaders, unreasonable schedule, etc. At the stage of software code development, there are risks: insufficient technical ability of developers, low development efficiency, problems in software testing, rework due to low software quality, new policies or new systems in the development process, etc. In the software delivery stage, the risks mainly include: customer dissatisfaction with the delivered software leads to rework; Construction period delay caused by test failure, demand change or other factors; The goal is not completed or the project fails due to other factors.

The harm degree of different risks at different development stages is also different. For example, if the problem of requirement change occurs in the requirement analysis stage, it will not bring great losses, but if it occurs in the development stage or even the test stage, it will have a great impact on the whole project. There are also some risk events that may occur throughout the software life cycle, and the impact on the software project is also random to a certain extent. For example, personnel changes may occur at any stage of a software project, and the impact on the project depends on the technical level of the personnel. The more core the member is, the easier the personnel transfer is to affect the project.

3. Risk Classification of Software Project

Software project risks can be divided into different categories according to different classification standards. According to the classification of risk sources, software risks can be divided into: policy risk, personnel risk, management risk, technology risk, demand risk, etc. According to the image classification of risk, it can be further divided into quality risk, cost risk, progress risk, etc. If the risk is caused by internal factors of the organization, it can be called internal risk, which is controllable by the development team itself and also the main management goal of risk management. The risks caused by external factors of the organization are called external risks. Common external risks include: project adjustment or stranding caused by policy changes, natural disasters affecting project progress, new technologies, more advanced purchasable parts, competitors using malicious means of competition, etc. Changes in customer requirements or excessive demands on software duration and quality are also external risks, but they can be controlled by sharing risks with customers. For example, if customers change their requirements in the development stage, they need to pay additional compensation and extend the construction period

according to the difficulty of modification. The external risks caused by the market or policies are often more difficult for the development team to control, and may bring much greater losses to the software project than the internal risks.

4. Steps of Software Project Risk Management

Many common software risks have been mentioned above, and it is not difficult to see that there is a certain randomness in the occurrence of software risk events. The purpose of software risk management is to reduce the probability of risk events and the losses caused. The software development team should establish a set of risk management process suitable for its own team, which should at least include the following steps:

Risk identification: understand the specific risk events that may affect the software project.

Risk assessment: analyze the software projects developed by ourselves, and sort the risks according to the probability of risk occurrence, the severity of harm to the whole development project and the priority of risk treatment.

Develop risk countermeasures: according to the risk assessment results, select the way to deal with risks, and appropriately modify the project plan to deal with risks.

Risk monitoring: throughout the project, continue to re-examine the risk profile, re evaluate the main risks, and update the risk profile according to the actions taken.

Risk identification and risk assessment should be carried out as soon as possible, so that risk management can obtain the most positive results in software projects. Risk identification involves collecting information about relevant software projects and classifying and summarizing them to determine the number of potential risks of the project. Analyze and evaluate risks to determine the possible impact of risks on the project. Formulate corresponding risk response measures based on the results of risk identification and risk assessment.

The risk management personnel shall select appropriate measures to deal with risks according to the actual situation. After the completion of risk identification and risk assessment, no matter whether the risk occurs or not, it is necessary to carry out risk monitoring so as to deal with possible sudden risks in time or adjust the software development plan appropriately in time. If the project team has the ability, it can also add the steps of recording and summarizing risk experience, recording the lessons learned in the previous risk management process or better risk handling means, and applying them in the subsequent risk management process.

5. Common Software Project Risk Management Methods

The above describes the basic steps of software risk management. Each step of risk management activities is relatively independent, so each step has different management methods. Different risk management methods have their own advantages and disadvantages [16]. The software

development team can choose appropriate methods to manage software risks according to its own actual situation and the specific content of the project.

5.1. Risk Identification Stage

In the risk identification stage, the common risk identification methods are as follows:

Brainstorming method: gather the project development team to hold a project risk identification meeting. The software risk manager explains the specific content of the software project in a simple and clear way, explains the rules of the meeting, and tries to create a relaxed and free atmosphere, so that the team members can discuss more "freely".

Checklist method: summarize the risk events or risk sources that have been experienced, and list a checklist of risk information that can be continuously used by the team. Each development can check and identify the possible risks from the list, and update the information in the table in time after each project is completed for the next risk identification.

Expert investigation method: in essence, it is a feedback anonymous inquiry method, also known as Delphi method. Ask the expert team for opinions on risk issues, collect, integrate, classify and sort the experts' opinions, and feed them back to the expert team again. Repeat the above steps until feasible opinions are obtained.

Flow chart analysis method: draw each link of software development into a flow chart, analyze each link in the flow chart one by one, find out the possible source of risk, and manage the risk in time.

5.2. Risk Assessment Stage

Common risk assessment methods are as follows:

Risk factor analysis method: the risk manager analyzes the factors that may cause the risk, looks for and investigates the source of the risk, so as to analyze the probability of the risk and the possible loss.

Qualitative risk analysis method: the priority of risk is obtained by analyzing and judging the causes of risk events, the impact of risk events on development progress, possible losses and other factors.

Probabilistic risk assessment model: its essence is a quantitative risk assessment method, which is mainly used for the risk assessment of complex systems. According to the probability of the root cause of the risk, the probability analysis method in mathematical statistics is applied to evaluate the probability of the risk event itself or its possible impact on the development project.

5.3. Risk Response Stage

There are mainly four types of risk response measures:

Reduce the probability of risk occurrence: according to the results of risk analysis and identification and risk assessment, adjust the project plan to reduce the probability of risk occurrence or minimize the impact of risk.

Risk avoidance: remove the modules or functions that may

cause risks from the development plan, and choose a safer or more robust way to achieve the target functions.

Sharing risks: sharing risks with users or customers to reduce their own losses. Before the start of the project, the scope of risk bearing shall be divided with the customer in advance to reduce their own losses.

Accept the risk: do not take any measures, and assume the loss caused by the risk if the risk occurs.

5.4. Risk Monitoring Stage

Risk reassessment: regularly reassess the risk of possible risk events.

Trend analysis method: through trend analysis of the frequency and time nodes of risk events in the software development projects that I have participated in, I can understand the high-risk nodes of different risks and monitor the software risks.

6. Suggestions on Functional Requirements of Software Project Risk Management System

The functional requirements of the software project risk management system are derived from the functions that the administrator may need in the process of software risk management. Suggestions for software project risk management system include:

6.1. Software Risk Item and Its Attribute Creation Function

The administrator creates a new software risk item and inputs the attributes of the risk item, including the risk name, software project ID, risk description, risk occurrence probability, risk severity and risk decision. The software risk management system will calculate the priority of the risk using the RAC risk assessment index method according to the risk probability and risk severity input by the administrator, And display the risk priority and other relevant information of the risk when the administrator inquires about the risk.

6.2. Software Risk Item Attribute Modification Function

Administrators can modify the attributes of risk items. With the development of the project schedule, various attributes of risks will change, and the means to deal with risks will also change. At different stages of software development, the priorities and coping methods of different risks will change. The software risk modification function can meet the needs of administrators to modify risks in time, improve the timeliness of risk management, and is more conducive to dealing with various uncertainties in the process of software development.

6.3. Software Risk Attribute Viewing Function

After entering the system, the administrator can view the existing risks, risk information and risk priority of different software projects. Since a software project has multiple risks, the search and query function can improve the efficiency of

managers. Search the risk name in the risk search bar to quickly query the risks that need to be managed.

7. Conclusion

Software risk refers to the uncertain factors or events in the software development life cycle that may lead to the loss of the development team or even the failure of the project. Software project risk has its characteristics different from the traditional project risk. First of all, the requirements of software projects are unstable compared with traditional projects. Customers often only have a general idea of the software they need. The software itself is the work of developers from the framework design to the final implementation. The requirements of customers change midway or there are misunderstandings between the development team and customers, resulting in requirements deviation from time to time. Secondly, human factors have a great impact on software development projects. Due to the rapid development of the IT industry, the speed of updating and iteration, and the working efficiency of experienced developers and novices is very different, it also has a great impact on the progress of the project. In addition, the results of software development are virtual software products without entities. Compared with traditional projects, it is more difficult to monitor the progress or quality of software projects. Therefore, risk monitoring of software projects is more difficult than traditional projects. In order to avoid risks and reduce losses caused by risks as much as possible, how to quickly identify and evaluate risks has become an important link in the software development process. The work done in this paper hopes to provide some reference for software project related workers, so as to better improve the success rate of software projects and avoid or reduce risks.

Acknowledgements

The paper would like to thank the support of the project "Research on New Generation Information Detection and Testing Technology and Methods" (40025001201838), as well as the authors of the references and relevant researchers, whose research has given me important reference and help, which has provided a good reference for the completion of my paper.

References

- [1] Wen Ruyun. Analysis of software project risk management [J]. Telecom World, 2022, 29 (3): 145-147.
- [2] Yang Hui. Research and Analysis of Risk Management and Control of Software Project [J]. Fujian Computer, 2021, 37 (5): 60-62.
- [3] Chu Hongjiang. Research on risk management strategy of IT software project [J]. China New Telecommunications, 2020 (19): 150-151.
- [4] ZhouYining, Chi Zhijie. Research and Discussion on software project management and risk assessment [J]. Network Security Technology & Application, 2022 (2): 63-64.

- [5] Pang Hongqiu. Research on Risk Management of Software Project Implementation in T Company [J]. Computer & Telecommunication, 2022 (1): 58-62.
- [6] Guo Jing, Wang Xiang, Zhang Hui, et al. Research on Risk Management of Software Development Project in M Company [J]. Science & Technology Information, 2021, 19 (9): 141-143, 147.
- [7] Xie Pengyu. Risk Management in the Software Development Process [J]. Management & Technology of SME, 2021 (5): 15-16.
- [8] Wen Linzhi. Dynamic risk management of software development based on life cycle [J]. Computer Knowledge and Technology, 2021, 17 (23): 78-79, 84.
- [9] Zhai Lanjie, Liu Zihao, Chen Yanhong, et al. On the role of risk management in software projects [J]. Chinese Information, 2021 (6): 266-267. DOI: 10.12221/j.issn.1003-9082.2021.06.266.
- [10] Yang Xiaojuan. Risk management process and risk decision analysis of large scientific research projects [J]. China New Telecommunications, 2021, 23 (7): 143-144.
- [11] Wang Fang. Analysis on project cost management of software development enterprises in China -- Taking Y company as an example [J]. Management and Administration, 2021 (3): 66-70.
- [12] Dou Youzhong. Risk control methods in software development [J]. Digitization user, 2021 (26): 136-138.
- [13] Zhu Hui. Thoughts on strengthening software engineering project management [J]. Kexue yu Xinxihua, 2022 (11): 34-36.
- [14] Liu Guanmei. Research on project management method of computer software engineering [J]. Sciences & Wealth, 2022, 14 (2): 31-33.
- [15] Liu Peng, Zhou Xia, Zhang Yi, et al. On software project management [J]. China CIO News, 2022 (4): 81-84.
- [16] Zhong Zhaoxu. Research on risk management mode of computer information system integration project [J]. China Broadband, 2022, 18 (7): 92-94.