Research on Risk Analysis and Management in the Software Development Process

Quanzhou Huang

School of Computer Science, Xi'an Shiyou University, Xi'an, Shaanxi, 710065, China

qzhuang@xsyu.edu.cn

Keywords: Risk analysis; Software process; Risk identification; Risk management; Unified software process

Abstract. Software risks are uncertain events in the process of software project management and development, and may bring harm or loss to the project plans and system quality. Project managers must identify the risk, analyze them and take action to manage the risks. This paper introduces the concepts of software risk, risk analysis and risk management. It discusses the processing methods of software risks and the means of estimating and managing risks. It mainly analyses the relationship of software risks and development process models. Risk management activities involve several important steps, including identification of the risks, analyzing them, and assigning priorities. It stresses the use of UML and UP process model have great role to reduce all the risks. Researching on software risk analysis and management has a great role for software production and software engineering.

Introduction

Software product is a logical product and produced by people through intellectual activities [1]. Software production is accompanied by risks and crisis [2]. Along with the increasingly complex software projects, software production and management are increasingly complex, software risks also will increase. The purpose of the study is to explore characteristic of software risks, risk treatment strategies, and techniques. Project managers must take steps to ensure that their projects are done on time and within effort and cost constrains. When unwelcome events may occur during development, managers make plans to avoid these events, minimize their negative consequences [4]. A risk is an unwanted event that has negative consequences [1,5]. Project managers must identify the risk, analyze them and take action to manage the risks. In this paper, starting from the concept of software risks, we study and explore the methods and means of software risk analysis and risk management.

Software Risks, Risk Analysis and Risk Management

Software development is a risky businesses, software risk hinders or delays the completion of tasks or projects. People usually think: software risks are uncertain events in the process of software project management and development, and may cause harm or loss to the project plans and system quality. A risk is an unwanted event that has negative consequences. Software risk can also be expressed in Eq.1.

$$\text{Risk} = \{ (\text{Ri}, \text{Li}, \text{Xi}) \} \qquad (1)$$

Among them, Risk represents a software risk set. R_i represents software risks, L_i represents the probability of occurrence of a risk, X_i represents risk results. The software risk actually is a collection of risks and its effects [6].

Software risk mainly has three different types: project management risks, technology risks and risks of product quality. The three kinds of risks are often linked together. There are a lot of risks in the software development process, some risks are hard to imagine, and difficult to estimate [7].

For software risks, we must carry on the risk analysis and risk management in software engineering activities. The process of risk identification, risk assessment and risk prediction is called risk analysis. The process of risk planning, risk control and risk regulation is referred to as risk management.

Software Risk Management Strategies, Risk Identification

Software Risk Management Strategies. Since the risks are unwanted events that have harmful effects on the system, so it must be taken seriously, and we must take effective risk management strategies. Like the deadlock problem in operating systems, software risks are also a kind of uncertain factors, and may or not happen. Thus risk management strategies have four kinds of methods: avoiding, restricting, reducing and monitoring [2,8].

Avoiding risks: by establishing a risk mitigation plan or changing the design, some risks can be avoided. Restricting risks: by constraints of some risks, that is, to make the risk impact only a small part of the system or programs. Reducing risks: when a risk occurs, we take some steps to alleviate the adverse consequences, and bottom out its negative effects. Monitoring risks: we should establish risk monitoring mechanism, for some unavoidable and unwanted events, to closely monitor and prevent it happening.

Software Risk Identification. Identification of software risks is trying to determine the various risk factors and events in the system, and predict its consequences. Identifying risks need to check all the documents in the system and all kinds of constraint conditions, and determine what special factors will threaten the project plans, what negative events might happen in the future, and where they might occur [9]. Managers can confirm the risks from the following several aspects: project risks,risks of costs and benefits, risks of development process, technology and environmental risks, the risks associated with the developers, and quality risks.

The managers analyze the risks they have identified, so that they can understand as much as possible about when, why, and where they might occur.

Risk Management Activities

Risk prediction table is established, and sorted by the endangering degree and emergency degree, and risk management plan of the project is put forward. According to the concrete conditions of each risk, we select one or more of strategies to deal with. In the process of software system development, from feasibility study, requirements, analysis, design, coding, testing, maintenance, to project management and quality assurance, each stage should be fully implemented risk management tasks. The risk management activities are shown in Fig. 1 [2].

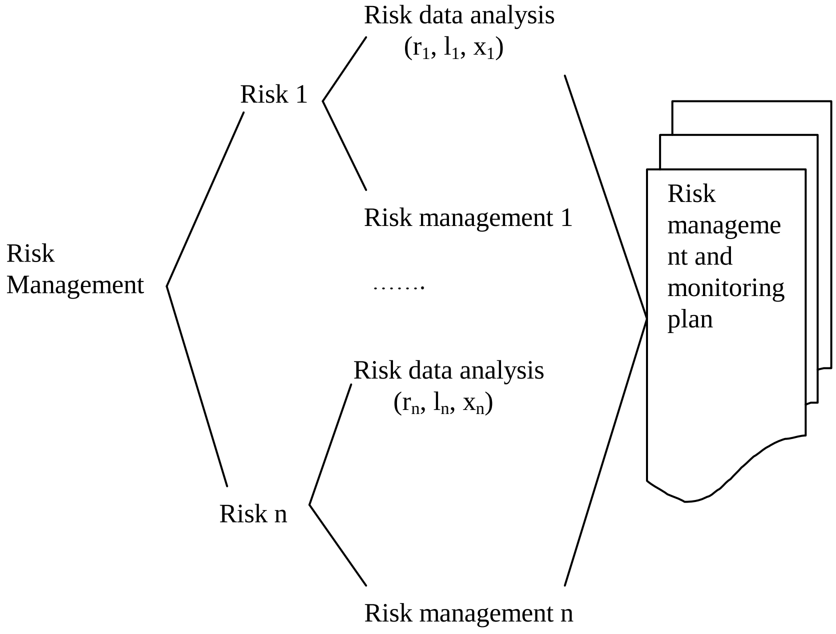

Figure. 1 Risk management activities

Relationship between Process Models and Software Risks

Waterfall Model and Software Risks. Waterfall model is divided into many stages: definition, requirements, analysis, design, implementation, testing, maintenance and so on. One development stage should be completed before the next begins. Users only see the final products and find the problems in the final stage. Thus, the risks are greater.

Prototyping Model and Software Risks. Through the establishment of software prototype, we can see the product in the early stages of the development. This method can improve understanding, when requirements are not easy to capture, but the lack of rigorous analysis and design will easily cause greater risks.

Evolving Process Model and Software Risks. After some stages, software products have been evolved. It is an iterative method, gradually formed perfect products. It includes increment model, spiral model, component assembly model and concurrent development model. Due to the lack of rigorous system analysis and design, these models have larger risks.

Object-oriented Method and Software Risks. Object-oriented method is the mainstream of development methods. It is starting from the relationship between object and object, trying to make problem domain and solution domain as consistent as possible, so as to find out a good software process model. The advantage of object-oriented method is consistent with the human way of thinking, the stability of developed products is good, the reusability of components is high, the maintainability of products is good, and the products are easy to test and debug. Overall, the object-oriented method effectively reduces the software risks, and is a kind of better development technology [10].

A Unified Software Process and Software Risks. Unified process is based on the components that have standard interface. The system is built and integrated through the connected components. The prominent characteristics of unified process are a use-case driven, architecture-centric, iterative and incremental development process. The unified process consists of several cycle, each cycle produces a product or version. Each cycle includes four phases: inception, elaboration, construction and transition. And each phase is a process of multiple iterations [3].

Iterative process is composed of some core workflows, mainly include: requirements, analysis, design, implementation, testing, etc. A typical iteration process includes all five kinds of workflows.. The relationship of each phase with the core workflows are shown in Fig. 2.

Because unified process uses unified modeling language (UML), life cycle has multiple cycles, the process adopts iterative and incremental development method, so it effectively reduces the risks of software process, it is also considered a risk-driven development methods.

How to Reduce the Risks for Software Development

It is almost impossible to completely avoid, prevent or eliminate risks. The question now is how to reduce the probability of occurrence of a risk and the corresponding consequences? In software engineering activity we should strengthen the research of risk analysis.

Software Engineering Methodology. In the process of software project development, the developers should follow the principles, methods, techniques and tools of software engineering, and should strengthen the system analysis and design, project management, and quality management. All these is very important to overcome the software risks.

Software System Modeling. Software system as a logical product, system quality is directly related to the integrity of the system modeling. System modeling and software risks are directly related. Unified modeling language (UML) should be adopted to strengthen the standardization of modeling, which is a guarantee of reducing the risks. Capturing the requirements and designing the system structure are very important to system quality.

Software Process Management. In order to effectively reduce the risks of software, we should strictly control and manage the software process, using the principles of unified software development process, use-case driven process, iterative and incremental development process, architecture-centric process and other new methods.

Software Quality Management. The ultimate goal of software engineering activities is to get high quality software products. Quality is related with the risks. In order to reduce the software risks the measures should be strengthening for project quality assurance. In each stage of development, we should review all plans, timely solve the problems, and prevent software defects or other bugs from passing to the next stage.

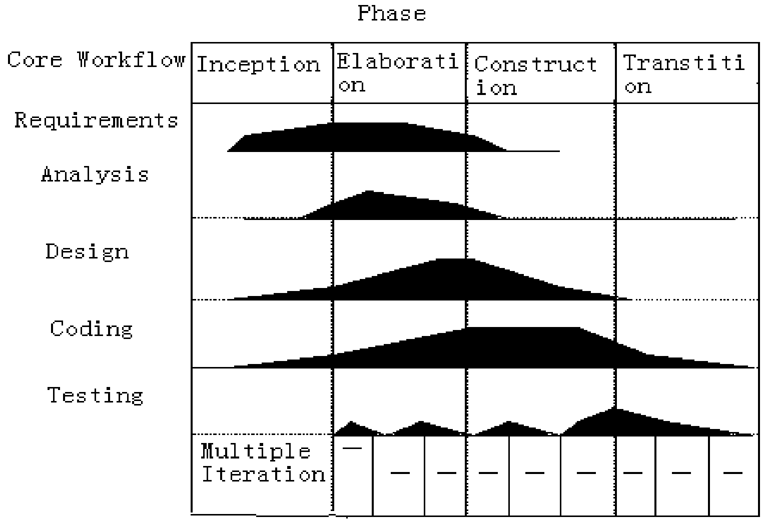

Figure. 2 Relationship of each phase with the core workflow

Summary

The software development process is accompanied by the risks. The risks will bring harm to the quality of projects and products. Therefore, risk management has very important significance. Risk prevention and risk analysis is an important task of the managers. This paper discusses the risk identification techniques, risk management activities and the risk management strategies. It mainly analyses the relationship of software risks and development models. It stresses the use of UML and UP process model have great role to reduce all the risks. Since the risks have a great impact on the project and product quality, so we will carefully study and plan solutions of the software risks in the development process, and explore all kinds of effective methods to control and manage it.

References

[1] R.S.Pressman:*Software Engineering: A Practitioner's Approach*(Mechanical Industry Press,China,2010),p.91. (In Chinese)

[2] S.L.Pfleeger and J.M.Atlee:*System Engineering Theory and Practice*(Higher Education Press,2012),p.100.

[3] I. Jacobson, The Unified Software Development Process, Mechanical Industry Press.2012. (In Chinese)

[4] Z.X.Tong and X.Liu: Journal of Heilongjiang University, Vol. 29 (2012) No.6,P.837. (In Chinese)

[5] N.Feng and M.Q.Li: Computer Engineering and Design, Vol. 27 (2006) No.7,P.1121. (In Chinese)

[6] Q.F.Yan, Z.Q.Nan and E.G.Yao: Computer Engineering and Design, Vol. 33 (2012) No.4,P.1581. (In Chinese)

[7] C.G.Pan and Y.W.Chen: Computer Engineering and Application, Vol. 32 (2014) No.32,P.9. (In Chinese)

[8] Y.Zheng: *Risk Management for Software Project Research and Application* (MS., Beijing Youdian University, China 2012), P.31. (In Chinese)

[9] Y.Zheng: *Research on Impact Modeling and Risk Management of Software Requirements Change* (Ph.D., Tianjing University, China 2011), P.25. (In Chinese)

[10] Y.H.Jiang and Y.P.Chai: Journal of Chifeng University, Vol. 31 (2015) No.2,P.114. (In Chinese)