Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2016, 3(6): 1-7

Review Article ISSN: 2394 - 658X

A Short Review for Selecting the Best Tools and Techniques to Perform Software Risk Management

Md. Forhad Rabbi[1] and Khan Olid Bin Mannan[2]

[1] *Roma Tre University, Rome, Italy*
[2] *Blekinge Institute of Technology, Sweden*
mrabbi@dia.uniroma3.it

ABSTRACT

The aim of this research paper is to study risk management system and to find some tools and techniques recommended by different journals and articles. We have gone through different approaches in context of risk management. We have taken risk management paradigm introduced by Software Engineering Institute as our standard to analyze different techniques and tools. Different features have been mined out from those models and trying to show shortcoming as well as asset qualities of those. Our approach is to find out best or suitable tools for software risk management in software development industries.

Key words: SEI, SRE, Softrisk, TRM, ARMOR, Riskit

INTRODUCTION

Richard E. Fairley defined 'risk' as 'the probability of incurring a loss or enduring a negative impact.' [1]. The Software Engineering Institute (SEI) defines risk as the possibility of suffering loss [2]. For this loss, an end product quality of a project could be decreased to an extent. There is a possibility, project cost and time could be increased and in the long run organizations can loss the market share. Like other businesses, software development organizations have a plenty of risks for their projects. So they also try to minimize risks and to ensure maximum profit [3]. Risk should be well controlled in the projects of software development organization as they also invest lots of resources on its development. So risk management has been introduced. It foresees the risks for the project and tries to understand those. Risk management technique draws a relationship of risks with project performance. According to Barry W. Boehm 'software risk management is an emerging discipline whose objectives are to identify, address, and eliminate software risk items before they become either threats to successful software operation or major source software rework' [4]. Nowadays a number of risk management tools and techniques have been established in software development industries, though these tools and techniques have benefits and few limitations too. Hence, we have decided to do our research by analysing and comparing these tools and techniques to enrich our professional skills and to give few recommendations on using of these tools and techniques. Software Engineering Institutes (SEI) proposed standard steps for overall risk management [2].

Our current study and analysis will be based on element those have been introduced in SEI software risk management paradigm. The rest of the paper is structured as follows. (i) Overview of current risk management models (ii) our research methodologies is mentioned (iii) details of the existing risk management techniques and tools is described. Finally our overall observation is discussed and ended with concluding remarks.

RELATED RESEARCH WORK

As part of our analysis we had to search for articles, journals, research works from different sources like IEEE, ACM, ELIN-BTH, Google etc. As a result we noticed that a number of researches have been done in the area of risk, risk analysis and risk management. What we have perceived the basic goal of all these research works was to find or to recommend more efficient tools and techniques to manage risk in an organized way in software development.

Nowadays several risk management techniques are being used in software developments industries. Risk management techniques get significant attentions because without managing risk properly very good project may fail. All of them has concentrated and tried to follow common procedure to manage risk through risk assessment and risk control. Time to time there has been revealed other risk management standard like SEI (Software Engineering Institute) and IEEE1540 standard.

There is no actual dominating risk management strategy. Different kinds of management approach have been introduced over time by different researchers. Software Risk Evaluation (SRE) [6] has been launched at around 1993 and second version of it has been released in 1999. This technique has been developed by SEI (Software Engineering Institute). SRE follow the SEI risk management paradigm. Through identity, analyse, plan, track, control and communication phase this model manage the risks of a project. Beside SRE Team Risk Management (TRM) technique has appeared by SEI/CMU team [7]. In TRM technique all stakeholders from developers to customers, all participate as team members in risk management process. Sometimes it seems good process because it ensures involvement of all individuals in project management. In the year of 2000 Softrisk [8], a risk management tool has been introduced by some researchers. This tool has focused on documentation and concentrates on top risks for the project by defining risk ranks. ARMOR (Analyser for Reducing Module Operational Risk) [9] automatically identifies risks of the project and maintain risk data repository. There is another risk management technique called Riskit [10]. Riskit is a systematic risk management approach. This technique use graphical formalism called Riskit analysis graph. A CMM based risk control optimization model also introduced as a technique in risk management areas.

RESEARCH METHODOLOGY

Among two research methodologies, quantitative strategies do complex experiments with numerous variables [5]. It has two different parts: experiments and surveys. Experiments include experiments and surveys. Inside qualitative method, case study is a method for learning about a complex example which comes through a complete understanding of that example. This method also includes few open-ended questions. When we started our study we realized after reviewing of two strategies qualitative method will suites perfectly with types of work, because our main goal is to review articles, case studies regarding risk analysis and management. Hence, finally we have decided to work with qualitative method for our research. The way we have designed our steps to move forward with this study is: 1) Firstly we will spend an specific time period to search for article and research papers and selection of most appropriate journals related to our analysis of risk management. 2) Analysis of collected results and recommendations from different research papers. 3) Comparison of different tools and techniques with standard risk management steps proposed by SEI. 4) Provide few recommendations on using of different tools and techniques in different software developments.

ANALYSIS OF AVAILABLE TOOLS AND TECHNIQUE

Software Risk Evaluation (SRE) Technique

The SRE is a software risk management technique which provides the most detailed outline. This approach concentrate on all the risk areas in the project in form of a classification based questionnaires [6]. SRE is not only a diagnostic but also decision-making tool for a project. This technique deals with risks from product, process and constraints. It identifies those risks and categorizes project risk statement. Moreover, project members take part in the risk identification and analysis. They try to mitigate risk areas facing their own development effort. This way project manager could be informed about the project risks at early stage. SRE introduces a set of risks managing activities. According to the Ray et al 'These risks managing activities can be integrated with existing methods and tools to enhance project management practices' [2].

SRE provides clear and understandable picture of the risks of the project. It diagnosis the risks and decide whether there any risk acceptable for staring a project or not. To identify risks before they become threat for project, SRE creates risk baseline. It can also reset risks baseline for the project. It works accordingly SEI standard and addresses the identification, analysis, planning, and communication elements of the SEI (Software Engineering Institute) Risk Paradigm. The analysis risk element is also covered fully by SRE activities. In SRE, construction of high-level mitigation strategy plans partially address planning element. The SRE also contributes significantly to the communication element.

SRE creates a shared view of risks facing a project among the staff, a common framework for talking about and mitigating risks. This risk management technique provides a snapshot of risks and enables the tracking of risks systematically (changes in probability and impact) and the tracking of risk mitigation efforts systematically. It also provides an impetus to focused project-level process improvement and decision-making information to the project manager and it accelerates the creation of a shared product vision among project staff.

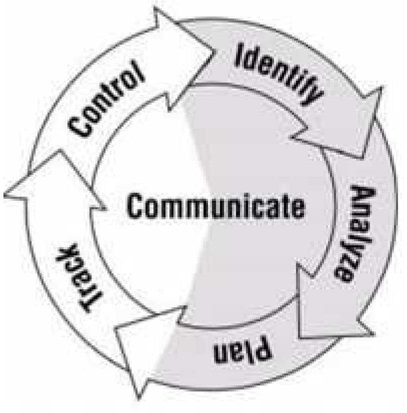

Fig. 1 SRE's Risk Management Paradigm [6]

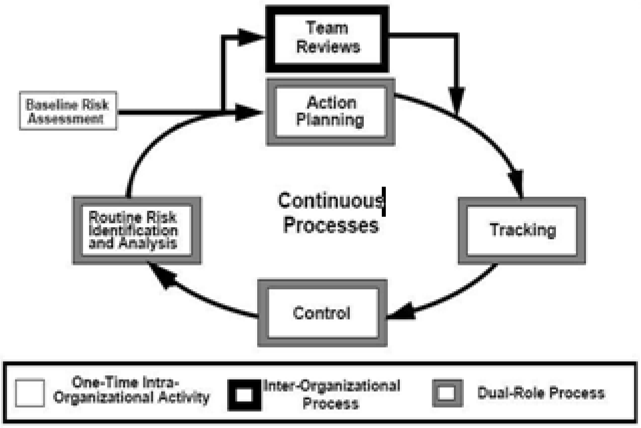
Team Reviews
Baseline Risk Assessment
Action Planning
Continuous Processes
Routine Risk Identification and Analysis
Tracking
Control
One-Time Intra-Organizational Activity
Inter-Organizational Process
Dual-Role Process

Fig. 2 Team Risk Management Process set [7]

Team Risk Management (TRM) Technique

According to the Higuera et al ‘Team Risk Management defines the organizational structure and operational activities for managing risks throughout all phases of the life-cycle of a software development program such that all individuals within the organizations, groups, departments, and agencies directly involved in the program are participating team members. Through the adoption of team risk management, the government and contractor are provided with processes, methods, and tools that enable both organizations, individually and jointly, to be increasingly anticipatory in decision- making processes’ [7]. Team risk management practices bring all individuals from developers to customers within an organization. This technique ensures continuous risk management throughout the project iteratively and cooperatively [13].

The processes of team risk management also address all five steps of the SEI paradigm through four processes. It combines the identification and analysis steps of SEI standards into the routine risk identification and analysis process. Risks not only exist in one stage but it can appear throughout the entire life cycle of a program. So this technique is a continuous process to identify and to control over risks. Here, a continuous cyclic set of scheduled activities are executed by team risk management for managing risks. The working procedure of TRM are identify risks, regularly review and analyse new risks, plan for sensible application of resources to ease risks, tracking of risks and risk normalizing actions, start controlling of risks that turn into problems, and finally start communication about risks among all partners in the program.

Softrisk Model

Softrisk [8] risk management technique has emerged because of the shortcoming of some traditional risk management technique. Softrisk model discusses risk management with other project management. This technique ensures risk automation and this model is appropriate for any type or any size of the project [14]. According to Keshlaf et al ‘The Softrisk model has been designed on the basis of the idea that risk documentation and concentrating on top risks are the best ways to save developers time and effort and produce good results in reducing software risks’[8]. The Fig. 3 has shown the Softrisk model ensures continuous management of risks till the end of the project. The model steps are briefly described as follows [8]:

The first step in Softrisk model is Risk Identification. Softrisk model identifies not only general kind of risk that can be occurred for any kind of project but also specific type of risk that only can be happened for some specific type of project. After risk identification in second stage of Softrisk model each identified risk is addressed in term of its probability and magnitude. Probability and magnitudes of the risks are estimated by a special checklist. Negligible, low, medium, high, very high and extra high are the categories under which risks are estimated. In risk documentation stage Softrisk documents all generic and specific risks data. This document is used for tracking projects situation, statistical operations and future risk predictions. Afterwards, risk assessment is done based on risk’s probability and magnitudes. Keshlaf et al defines risk exposure formula:

RE= Risk probability* Risk Magnitude [8]

In fifth stage, RE valued are used to sort all risks and top ten risks for each inspection prioritized and listed. Then a graph is used to represent RE values by dividing three zones red, yellow and green. In controlling phase a suitable reduction technique is chosen based on severity of the risk. This technique could be mitigation, contingency or crisis plan. After using any of this reduction technique re-assessment, re-estimation and re-documentation is required. And in the last stage of Softrisk model to be confirmed about nonexistence of any risk, this approach can be stared from began.

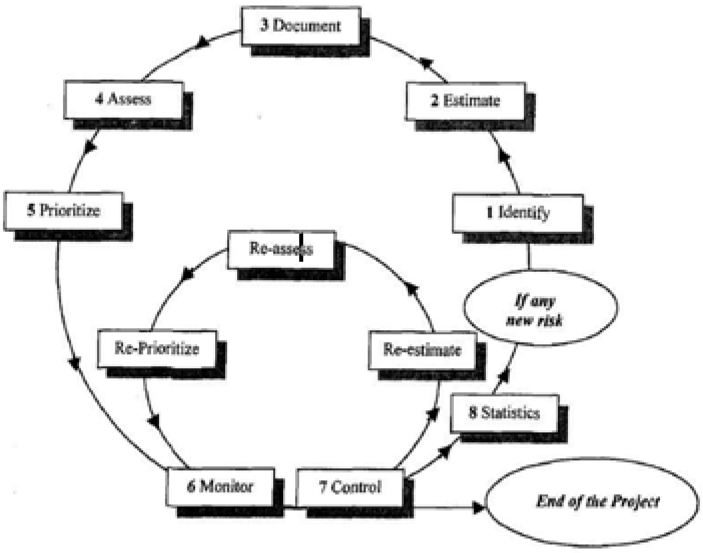

Fig. 3 Softrisk Model's diagram [8]

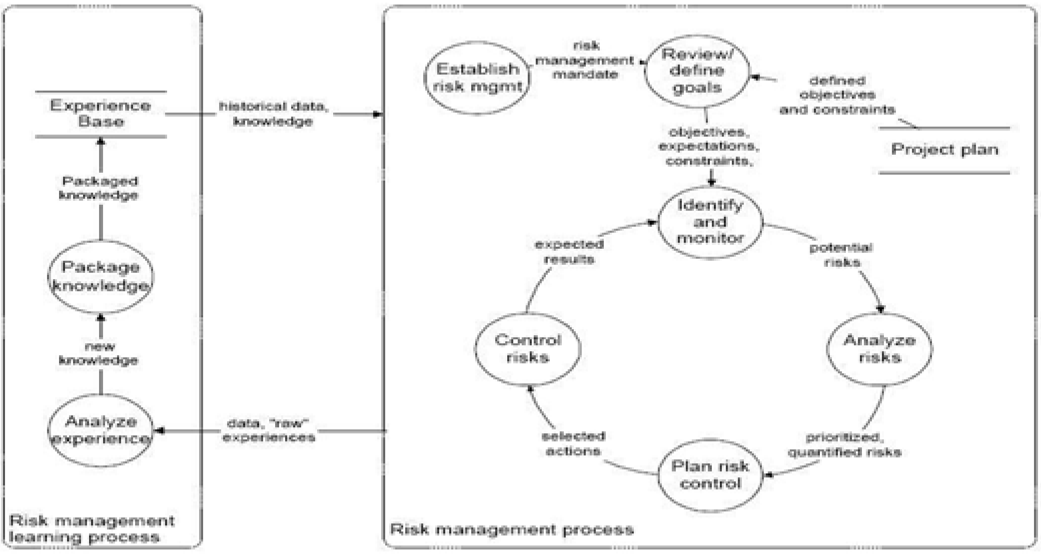

Fig. 4 Riskit risk management cycle [10]

Analyser for Reducing Module Operational Risk (ARMOR)

According to Lu et al ARMOR (Analyser for Reducing Module Operational Risk) is a software risk analysis tool which automatically identifies the operational risks of software program modules' [9]. ARMOR's working procedure is to access data repository, project database, failure database and program development database. This tool create risk model and display various statistical quantities for project management. Risk modelling procedure is being simplified by enhancing user interface. ARMOR establishes promising risk models for the project under evaluation and can measure the risks of software programs within the project. This tool can perform to identify the source of risks and can also designate how to improve software programs to reduce their risk levels. The validity of risk models from field data is determined by this tool.

In ARMOR users can apply software metrics to create various risk models in addition to compute module risks. After creating and executing risk model the predicted risk of each module is displayed. Various statistics on the distribution of software risks can be displayed by the users in ARMOR tool. Finally, to determine the validity of the risk models users apply regression analysis. This validated risk models are usually saved in a model repository. This model would be used in further applications to another project release or a completely different project.

Riskit Model

Riskit [10] is a risk management method that has been developed to provide a theoretically sound as well as a practical risk management approach. The method used in a number of industrial projects in Europe and USA. The reason behind the developing of Riskit is to support systematic risk analysis. An effort has given to design the method preparing a goal of avoiding hazard section. Riskit approach supports qualitative analysis of risk situation.

A graphical formalism is used by this tool. This technique can rank the risks, based on the availability of historical data, on accurate assessment and on utility theory. One feature of this tool is, it supports multiple goals and stakeholders. The Riskit analysis graph is a graphical formalism in Riskit model to document risk. Using the Riskit analysis graph different aspects of risk explicitly are defined. Hence it is more formal than face to face communication. The Riskit analysis graph decomposes risks into risk elements during the Riskit process.

According to the Fig. 4 risk management infrastructures defines method, techniques, responsibilities and the scope of risk management in Riskit management cycle [11]. It reviews the predefined goals for the project, refining them and also classifies implicit goals and constraints explicitly. Riskit model deals with all stakeholders of the project and with their associated goals. According to the Riskit management cycle, it identifies risks and monitor in risk identification and monitoring phase. In risk analysis phase it pigeonholes identified risks into risk factor and risk events. Then it accomplishes risk scenarios for all risk events those have been classified in risk analysis phase and estimates risk effects for all risk scenarios. Afterwards, suitable levels of metrics are used to estimate probabilities and utility losses of risk scenarios. In risk control planning phase, a ranking of risk scenarios have made based on their probability and utility loss for each stakeholder. Controlling of risks in Riskit risk management, implements the risk controlling actions.

CMM based Software Risk Control Optimization Model

In CMM based model, software risk assessment and control follow the CMM (Capability Maturity Model) framework. In this model process database is used to identify risks and corresponding mitigation steps [15]. CMM (Capability Maturity Model) based software risk control optimization model implemented a software risk control policy based on the historical data of similar projects [12]. In this model software risk management process works combine into two activities: software risk assessment and software risk control. Software risk assessment is a discovery procedure of identifying sources of software risk, to find out their potential effects and finally to prioritize them. Software risk control is a process of preparing risk resolution plans, monitoring risk status, implementation of risk resolution plans and correction of derailed from the plan. When the identification and prioritization of the risks are complete, a well-structured planning is necessary to minimize the consequences of top risks in the list. In this model a process database or risk repository plays a vital role to identify and generate risk control decisions.

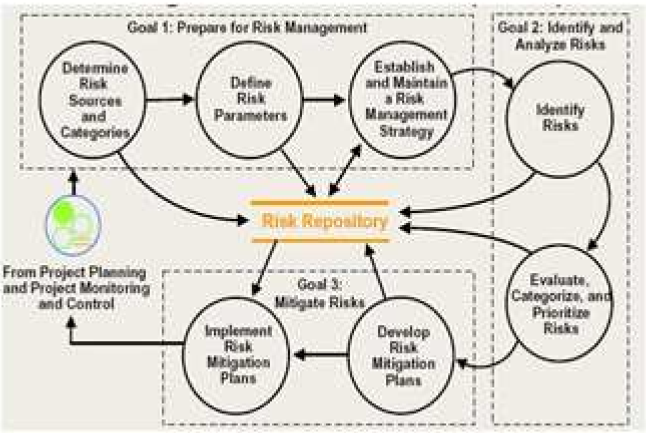
Goal 1: Prepare for Risk Management
Goal 2: Identify and Analyze Risks
Determine Risk Sources and Categories
Define Risk Parameters
Establish and Maintain a Risk Management Strategy
Identify Risks
Risk Repository
From Project Planning and Project Monitoring and Control
Goal 3: Mitigate Risks
Evaluate, Categorize, and Prioritize Risks
Implement Risk Mitigation Plans
Develop Risk Mitigation Plans

Fig. 5 CMM Risk Management Processes [12]

SEI STANDARDS

Risk management is a systematic and continuous process. SEI (Software Engineering Institute) risk management paradigm can describe this best. According to SEI [2] there are seven elements in risk management paradigm. Those are risk identify, analyse, plan, track, control and communication. These risk management steps occur sequentially. But these activities happen continuously, concurrently and iteratively. When new risk is identified and analysed at same time other risks are tracked. In Fig. 1 there is shown SEI risk management paradigm. SEI thinks all risk management technique will follow their prescribed paradigm for risk management of software project.

In identify phase all known project risks should be explicated before they become threat for the project. In this step one risk management technique should identify risk before starting the software project.

In analyse step of SEI paradigm risk data should be renovated into decision-making information. In this steps after identifying risk one technique should investigate the risk information to find out which are threaten for project and to sort them according to priority.

In planning step, in SEI risk management paradigm, risk information should be transformed into decisions and mitigating actions. And afterwards it implements those actions. According to the plan risk should be alleviated by the risk management technique.

Risk indicators and mitigation actions are monitored in track phase of SEI risk management paradigm. This step is also important like other steps. Without proper tracing risks may arise later.

Control in SEI risk management paradigm, keeps monitoring and corrects the deviations from the risk mitigation plans. In any stage of the software project risk management technique can be divergence from its main track. So risk management technique should correct it.

In communicate stage; all information should be shared throughout the project. All stakeholders should be well informed about project risk, mitigation plan and controlling.

SEI risk management standards ensure proper risk management through these seven stages.

DISCUSSION

Software risk management is an emerging discipline whose objectives are to identify, address, and eliminate software risk items before they become either threat to successful software operation or major sources of software rework. Software Risk Evaluation (SRE) technique is a good example of risk management approach. It identifies risk, analysis it and makes plan according to the analysis. Identification and analysis phase are conducted by only project personnel, no involvement of other individuals or stakeholders. This could be exigent issue for this model. Because major risks for the project are raised from requirement factor and requirements are provided by stakeholders. So lack of stakeholder participation, project development team may not get clear requirement for the project that can lead them to failure. On the other hand TRM ensures all individuals participation in risk management process. And it helps the project manager to get the correct requirement of the project. In TRM, risk management practices fetch individuals together within an organization. It has intra organization activity and inter-organization process.

Softrisk approach concentrates on documentation. This model can be used by any type of project, in any phase of project. An important feature of this tool is it can measure magnitudes and probability of the risks. It sorts out top risks among all and emphasis on those risks to mitigate. It uses graph that ease the risk monitoring. It introduces risk reduction advice in terms of risk mitigation plan. Softrisk uses risk data repository as like ARMOR does. ARMOR is a systematic approach that not only identifies risk but also identifies source of the risk and takes initiative to improve the project by reducing risk level.

Riskit management model is popular in many industries. It uses graph to define different aspect of risks. It also recognizes all the stakeholders who have interest in the projects. It ensures the main features of TRM participation of stakeholders as well as highlighting the important risk which is the main characteristics of Softrisk.

In CMM based software risk control optimization model approach, we have found a similarity with ARMOR tools in their working procedures. They both work collecting data from previous failure databases which is using of heuristic knowledge. This model’s basic steps are based on SEI standards which we have followed all through our study.

CONCLUSION

The plan we had set to do a study on popular risk management tools and techniques for software development, we gave our full effort to execute the plan through a study of a good numbers of journals and research papers. Our intention was to find out better risk management tools and techniques by showing a comparative analysis. At the end of study, our comparative analysis shown that no single tool or technique stand alone is perfect for managing risks in software development because of their different features and working procedures. So, we believe our comparative analysis will help and to provide basic ideas to the organizations in selection of their risk management techniques depending on their developments.

REFERENCES

[1] Richard E Fairley, Software Risk Management Software, IEEE Software, California State University, June, **2005**, 22, (3), 101 – 101

[2] Ray C Williams, George J Pandelios and Sandra G Behrens, *Evaluation (SRE) Method Description* (Version 2.0), CMU/SEI-99-TR-029, ESC-TR-99-029, **1999**.

[3] He´lio R Costa, Marcio de O Barros and Guilherme H Travassos, Evaluating Software Project Portfolio Risks, Journal of Systems and Software 80:16-31 · Rio de Janeiro, Brazil , January 2007

[4] Barry W Boehm, Software Risk Management, *IEEE Computer Society Press*, Piscataway, NJ, USA **1993**,

[5] W Creswell, *Research Design - Qualitative, Quantitative and Mixed Method Approaches*, Sage Publications, **2002**, 13-17.

[6] Carr M.J., Suresh L., Konda Ira, Monarch F., Carol Ulrich Clay F. Walker *Taxonomy-Based Risk Identification*, SEI Technical Report CMU/SEI-93-TR-006 ESC-TR-93-183, SEI/CMU, Pittsburg, PA, **1993**.

[7] Ronald P Higuera, David P Gluch and Richard L Murphy, *An Introduction to Team Risk Management*, Special Report CMU/SEI, SEI/CMU, Pittsburg, PA, **1994**.

[8] Ayad Ali Keshlaf and Khairuddin Hashim, A Model and Prototype Tool to Manage Software Risks APAQS '00 Proceedings of the The First Asia-Pacific Conference on Quality Software (APAQS'00) Page 297 IEEE Computer Society Washington, DC, USA ©2000

[9] MR Lu, JS Yu and SR Dalal, ARMOR: Analyzer for Reducing Module Operational Risk, *Twenty-Fifth International Symposium on Fault-Tolerant Computing*, Pasadena, CA, USA, **1995**, pages: 137 - 142.

[10] Jyrki Kontio and Victor R Basili, Empirical Evaluation of a Risk Management Method, *SEI Conference on Risk Management*, Atlantic City, NJ, **1997**.

[11] J Kontio, Risk Management in Software Development: A Technology Overview and the Riskit Method, *Proceedings of the International Conference of Software Engineering*, Los Angeles CA, **1999**, 679-680.

[12] Xu Ruzhi, Qian Leqiu, Jing Xinhai, CMM based Software Risk Control Optimization, *IEEE Journal Information Reuse and Integration*, **2003**, 499-503.

[13] David P Gluch, Audrey J Dorofee, Elizabeth A Hubbard and John J Travalent, *An Collaboration in Implementing Team Risk Management*, Technical Report, CMU/SEI-95-TR-016, **1996**.

[14] L Jamie, Shawn A Smith, D Bohner and McCrickard Scott, Project Management for the 21st Century: Supporting Collaborative Design through Risk Analysis, *43rd ACM Southeast Conference*, USA, **2005**.

[15] Pankaj Jalote, *CMM in Practice: Process for Executing Sofmare Projects at Infosys*, Addison- Wesley, **2000**, 5-117.