Pioneering an Automated Risk Removal Tools in Software Engineering

Dr. M. M. Sharma
Principal, Govt. Engineering College,Ajmer

Anil K. Dubey
Govt. Engineering College, Ajmer

Prakriti Trivedi
Asst. Professor, Govt. Engineering College, Ajmer

Akanksha Toshniwal, Himanshu Swarnkar
Govt. Engineering College, Ajmer

***Abstract*—Due to the dynamic changes of business environments and the advancements of technologies, information technology (IT) projects are facing lots of challenges, and there is requirement of systematic approaches to deal with the risks to ensure the project's success. As, no IT Project can ever be risk free. Here we are proposing a framework of automated risk removal model which identifies, classifies and generates solution patches to solve the problems caused by software risks. This is a phase wise process of risk removal. First, risk sources and risks are identified, after that they are classified, and then solution patches are generated and applied on encountered problem. At the end performance of the project is evaluated after risk removal. This framework of process has auto feedback mechanism. Using this model framework project manager will be able to develop better risk management strategies and more effective risk planning decisions.**

Keywords—Risk sources, risk identification, project risk management, risk measurement, solution patches.

I. INTRODUCTION

As no IT project can ever be risk free, many methodologies have been applied to enumerate the possibility and estimate the impact of risks that a project may encounter.

Software risk management is the figurative process, in which risk factors are systematically identified, assessed, and mitigated. To find out the software risks in a project either due to external or internal causes is a major part of project management. A subset of project management includes the processes concerned with identifying, analyzing and responding to project risks.

We introduce an outline of set of novel practices to identify risks and risk sources, classify them and applying auto feedback solution patches to deal with the problem caused by the software risks. Learning from past experiences that, what resulted into the software failure helps to make this model framework better. Risk management continues throughout the life cycle until the product is delivered [1]. Fig. 1 shows how risk management fits within the software project management life cycle.

The paper is organized as follows: Section II summarizes the previous work in risk management and risk assessment. Section III describes the model framework of automated risk removal model (ARRM), its algorithm and flowchart. Section IV includes the importance of ARRM and offers our conclusion and an outline of future research.

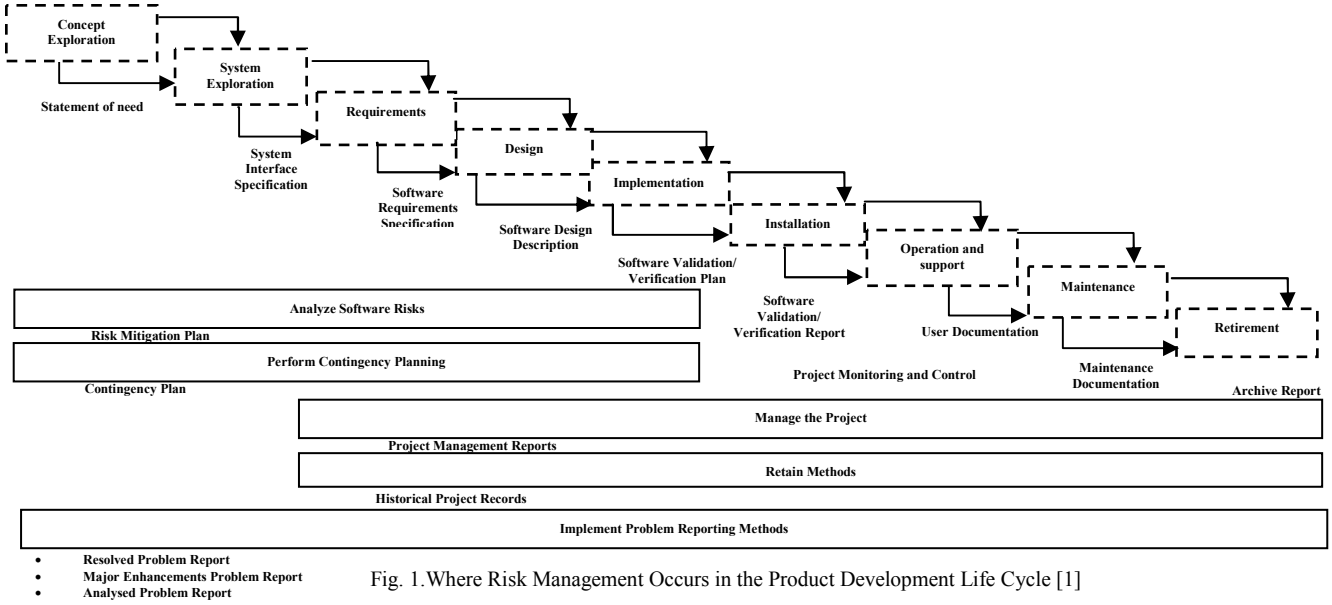


Fig. 1.Where Risk Management Occurs in the Product Development Life Cycle [1]

978-1-4673-5986-3/13/$31.00 ©2013 IEEE

II. RISK MANAGEMENT REVIEW

The Software Engineering Institute (SEI) [2] has developed a risk management paradigm, which is an expansion of the classic "plan-do-check-act" cycle and states a set of cyclic steps (i.e., Identify, Analyse, Plan, Track, and Control) throughout the project. It underlines the risk management as a continuous process in which each risk goes through these steps sequentially and independently.

The common risk management processes and management practices accepted in the industry can be mapped to the SEI paradigm. Risk Management Practices [3] consider each of these five practices in turn.

A. Identify Project Risk

Current risk identification processes involve exploring the major areas of a project, collecting input from personnel, learning from past experience, and applying analytical tools and techniques [1], [5], [6], [7].

Most of these approaches identify and manage events independently and tend to identify risks rather than opportunities; so, these approaches are often complemented with techniques such as SWOT (Strength, Weakness, Opportunity, Threats) analysis, constraints and assumptions analysis and forcefield analysis [8].

B. Evaluate and Prioritize Risk

There are two major types of risk evaluation in project management, Qualitative Risk Analysis and Quantitative Risk Analysis [8], [9]. The methods commonly applied in the project risk analysis are Failure Mode Effect Analysis (FMEA) and Failure Mode Effect and Criticality Analysis (FMECA). FMEA is used to identify the risks and their associated effects and FMECA [10] is used to rank the risks according to their criticality and their probability. Risks are usually ranked with assigned relative scale values to Probability and Impact of the risks.

C. Develop Risk Response Plans

The table based risk ranking approach allows organizations to select appropriate risk response strategies. Table 1 [12] summarizes the typical risk response actions based on the evaluated result.

D. Monitor Status of Risk and Associated Risk

Risk monitoring is carried out continuously throughout the project life cycle. The main objective is to monitor any changes of identified risks, the effectiveness of risk responses and the performance of the implementation of risk management practices [5], [9], [11].

E. Control Risk Response Actions

Risk control is also an ongoing process for the life of a project. With the risk monitoring results, risk control involves reassessing risks and selecting alternative risk response actions [5], [9]. As there may be status changes of existing risks, new risks identified, or variances of planned against implemented responses, all risks have to be re-evaluated and reprioritized periodically so that suitable decisions and risk responses can be made. Based on the risk re-evaluation and reprioritization results, the risk response plans should be reviewed and updated.

III. AUTOMATED RISK REMOVAL

Current project management practices do not clearly address where the risk is coming from and how the problems caused by the risks can be taken to the solution. Techniques which are available do not provide automated software risk removal. Automated risk removal is novel method in Fig.3 gives different phases of risk removal. In this section, we are proposing an algorithm for risk removal. It starts with the first phase of identification of software risks and its sources, then they are identified and analyzed before going into next phase, once the risks are identified and analyzed and then each risk is classified as Low Risk (L), Medium Low Risk (M-), Medium Risk (M), Medium High Risk (M+), and High Risk (H), and then auto feedback solution patches are generated to get the solution to the problem caused by the risks and after that, the patches are applied to the problem occurring in the project.

TABLE 1. RISK RESPONSE ACTIONS

Severity Level	Probability	Impact	Purpose of Response Actions	Description
Risk (I>0)				
High	High	High	Reduce the impact and probability	Action taken to reduce severity level or remove the risk
Medium	High	Low	Reduce Probability	Action taken to reduce the risk likelihood
Medium	Low	High	Reduce Impact	Action taken to reduce risk effect
Low	Low	Low	Monitor Risk	No actions to risk, except monitoring
Opportunity (I<0)				
High	High	High	Exploit Opportunity	Actions taken to realize Opportunity
Medium	High	Low	Enhance Impact	Action taken to enhance positive effect of the Opportunity
Medium	Low	High	Enhance Probability	Action taken to enhance likelihood of Opportunity
Low	Low	Low	Ignore Opportunity	No actions are needed

A. Identify Risk and Risk Sources

There are basically two methods for risk identification. First is talking to people, and another is reading documents. First method includes interview, brainstorming, and consulting experts. Second method includes study project documentation, review planning, study specialist literature, checklists, research assumption, and many more. Identification of risk

sources [13] provides a basis for systematically examining changing situations over time to uncover circumstances that impact the ability of the project to meet its objectives. Risk sources are both internal and external to the project. As the project progresses, additional sources of risks may be identified. Establishing categories for risks provides a mechanism for collecting and organizing risks as well as ensuring appropriate study and management attention for those risks that can have more serious consequences on meeting project objectives. Risk analysis is a systematic approach for describing and/or calculating risk. Risk analysis involves the identification of undesired (accidental) event, and the causes and consequences of these events.

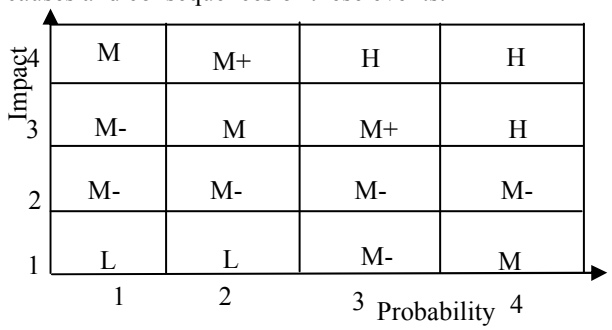


Fig. 2. Risk Assessment Matrix

B. Risk Classification and Prioritization

Once identified, all risks were initially estimated and added into a centralized risk record. The estimations were performed by the risk originator (i.e. one who identified the risks) using a Ranking method, which was based on the different combination of assigned probability and impact values of their risks. According to the predefined risk assessment matrix [12] used by the organization, each risk was measured by assigning appropriate probably and impact value, both ranging between 1 and 4, where higher probability value representing a higher chance that the risk would occur, and higher impact value representing negative effect that the risk would impact the project.

Based on the different combination of assigned probability and impact values, a risk's severity level and its response priority were determined before any response actions could be planned. According to the predefined risk assessment matrix used by the organization as shown in Fig. 2 [15], each risk is then classified as Low Risk (L), Medium Low Risk (M-), Medium Risk (M), Medium High Risk (M+), and High Risk (H); in addition, as for calculating risk scores, each risk classification was further assigned a pre-defined score from 1 (low risk) to 4 (high risk).

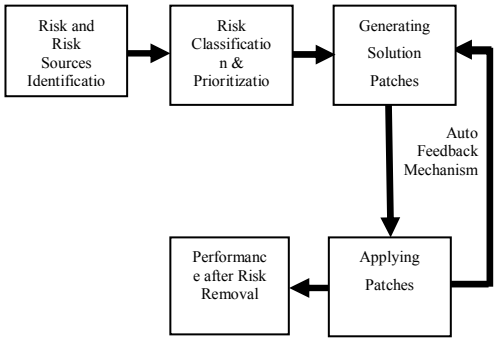


Fig. 3. Automated Risk Removal Framework

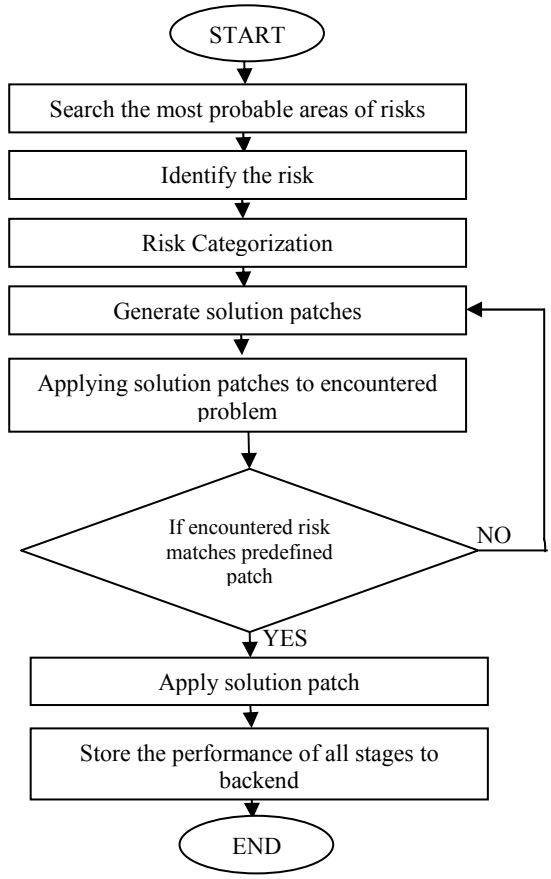
C. Generating Solution Patches and Applying Patches on Problem

This is the empirical state works on the basis of auto feedback mechanism. This state works in two phases, first phase of generating new solution patches (small procedures that are applied on the problem encountered by the risk in the project to get the solution of the problem) to solve the problem encountered. It includes two states, first where the solution patches are generated, and second where solution patches are applied and stored for future use. This state is based on the practice of learning from the past experiences. Auto feedback mechanism refers to the process of sending back the flow to generate patch, if the solution patch is not found in the backend database. First time when the problem is encountered new solution patch is generated and stored at backend database, so in future when similar problem arises these patches are applied to deal with it, this reduces the cost of every time creating solution patches and enhances the performance.

D. Performance after Risk Removal

When the solution patches are generated and applied to the problems. The problem occurs due to risks can be removed effectively. In this stage the performance of all steps of risk removal is stored at backend database. Once the risk is removed the performance is measured and then decision can be taken on the basis of performance measure whether to go further with the project or not.

IV. AUTOMATED RISK REMOVAL FLOWCHART



V. AUTOMATED RISK REMOVAL ALGORITHM

Start

1. Get the project
2. Identify the risk and its sources
   - 2.1 Concern people and collect primary data
   - 2.2 Manage and compile secondary data
3. Risk classification(According to Risk severity level)

   Priority based risk classification, each risk is classified as:-
   - Low Risk (L)
   - Medium Low Risk (M-)
   - Medium Risk (M)
   - Medium High Risk (M+)
   - High Risk (H)

4. Generating solution patches,

   According to predefined rules and regulations the patches are designed. It contains instructions and schedule fundamentals to solve similar types of problem. These are known as Instruction Set to solve the problem.
5. Applying solution patches
   - 5.1 If (the encountered risk matches the predefined patch)

     then, apply matched patch to remove the risk.
   - 5.2 Else

     Go to Step 4.
6. The performance of all steps of risk removal is stored at backend database.

End

VI. CONCLUSIONS AND FUTURE WORK

We have proposed a model framework of automated risk removal. First of all, we gave a brief of how risk management fits within the software project management life cycle. In this model framework, we gave an outline for applying different methods to identify risk and its sources, to analyze risks and to measure it by setting its priority called prioritization. We used auto feedback mechanism to deal with the problems occurring due to risks and every time we stored the solution for future use. This framework model moves forward on the concept of learning from the past experiences, where project managers faced software failure and learnt what practices prevent from project failure.

At last, the framework enhances the performance of the project after risk removal.

Besides, the proposed framework has a number of areas that need further verification and improvement. We will work in the efforts for further improvement of this framework.

- As the current practices of this framework ignore the existence of risk dependencies, another area of research is to apply the risk dependency concept to this process.
- As the risk is dependent on each other, what if solution applied to one risk results in unfavorable consequence on other risks. Further work is required to study how this framework can relate to risk dependency.
- This framework can be included along with the spiral model.

We will work on the improvement and enhancement of this framework.

ACKNOWLEDGMENT

I would like to thank Dr. M. M. Sharma, Prakriti Trivedi, Anil Kumar Dubey, along with the anonymous references for their helpful study matter.

REFERENCES

[1] Don Shafer, "Software Risk: Why must we keep learning from experience," Athens Group, Inc., Houston, Texas, USA, 2004.

[2] R. Van Scoy, Software Development Risk: Opportunity, Not Problem. Software Eng. Inst., Carnegie Mellon Univ., 1992.

[3] T. Kwan and H. Leung, "Improving Risk Management Practices for IT Projects," Proc. IASTED Int'l Conf. Advances in Computer Science and Technology,2007.

[4] Project Management Inst.,A Guide to the Project Management Body of Knowledge (PMBOK Guide),fourth ed. Project Management Inst., 2008.

[5] Software Eng. Inst., CMMI for Development, Version 1.2,Carnegie Mellon Software Eng. Inst., Aug. 2006.

[6] Am. Systems Corporation, Risk Management Process and Implementation .Am. Systems Corp., Chantilly, Va., 2003.

[7] D.Hillson, "Extending the Risk Process to Manage Opportunities,"Proc. Fourth European Project Management Conf., June 2001.

[8] COSO (Committee of Sponsoring Organisations of the Treadway Commission),Enterprise Risk Management: Integrated Framework, www.coso.org/publications.htm, 2004.

[9] Project Management Inst.,A Guide to the Project Management Body of Knowledge (PMBOK Guide),fourth ed. Project Management Inst., 2008.

[10] D. Lock,The Essentials of Project Management, third ed. Gower Publishing, 2007.

[11] IT Governance Inst., IT Assurance Guide Using COBIT, The IT Governance Inst., 2007.

[12] Tak Wah Kwan and Hareton, " A Risk Management Methodology for Project Risk Dependencies," IEEE Transactions on Software Engineering, VOL. 37, NO. 5, 2011.

[13] Warren Scheini, "A Comprehensive Survey of Risk Sources and Categories," NDIA 4thAnnual CMMI Technology Conference & User Group November, 2004.

[14] Su-xian ZHANG, Rui-li YANG, "A Dynamic Model of construction Project Risk Measurement Based on the Process," 978-1-4244-6484-5/10/IEEE, 2010.

[15] Kwan Tak WAh, "Risk Management Methodology for Risk Dependencies," Hong Kong Polytechnic University, 2009.