A New Model for Software Risk Management

Bharat Singh*, Kapil Dev Sharma2, Subhash Chandra3
1Central University of Rajasthan, 2Anand International College of Engineering, 3RIET, Jaipur
1* bharatkhatana@gmail.com
2shkapil@yahoo.com
3subhashccjat@yahoo.com

Abstract: In the modern time of software engineering, the development of software in static and dynamic environment results in several vulnerabilities that need to be handled so that they do not step in with the clear defined project goals. Previous studies show that the wide variety of different risk analysis strategies provide a valid solution to address the lack of risk management strategies in Software risk assessment model (SRAM), Software risk assessment and estimation model (SRAEM) etc. In this paper we proposed a new model (SRAIEM) software risk assessment identification and estimation model for the software risk management. This paper focus on take care of most frequently occurring risks and it is possible to predict the possible result of software projects with good accuracy proposed model not only access the risk but also identify and estimate the risk. This model provides a way to managing all of them. In order to cover some perspectives of the software industry which have not been covered up to now.

Keywords: Software, Risk estimation, Risk management, SRAIEM.

Introduction

Risk management is a action that help a software team to understand and manage uncertainty. Risk management also partly means reducing uncertainty. A risk is a potential problem – it might happen or it might not. But regardless of the outcome, it’s a really good idea to identify the risk, assess its probability of occurrence, and estimate its impact. Everyone involved in the software development projects i.e. software engineers, managers and other stakeholders must participate in risk analysis and management. But software risks have been increasing as long as the software industry has been growing. Many software development projects miss their goals of delivering a acceptable software product due to poor time management and budget and poor SRM. Many software engineers and project managers have only a limited understanding of the concepts of SRM, due to which they can’t produce a expectable outcome.

When risks are analysed, it is important to measure the level of uncertainty and the degree of loss associated with each risk. To accomplish this, three dimensions of risk are considered. (i) Project Risk (ii) Technical Risk (iii) Business Risk. The project risks threaten the project plan. That is, if the project risks become real, it is likely that project schedule will slip and as a result cost will also increase. Technical Risk threatens the quality and timeliness of software to be produced. Business Risk threatens the normal growth and development of software to be built and often threaten the product or project. It has been noticed that that this simple risk categorization won’t always work. Some risks are simply unpredictable in advance. In spite of this, not all organizations or software development projects uses these procedures and even if they do, they cannot use them successfully. The paper is organized as follows. In section-II we present the background and related work of risk management and various risk assessment models. In section-III we present the comparison of different software risk assessment models. In section IV we have presented the comparison of different software risk assessment models corresponding to risk factors. In section V we have done the results and analysis. Finally we have concluded the work.
II. Background and Related Work

Software risk management is an organized way of identifying risk factors, analysing them, assessing their impacts on the software development projects and mitigating them when they arise. Systematic risk management may be facilitated through the use of wide variety of risk assessment models that provides quantitative assessments of the risks involved. There are few published risk assessment models that evaluate the risk of software projects.

In a model has been developed in which risk is estimated using software metrics of risk management. This model gives the incremental risk for every phase and also the total cumulative risk as the software progress from phase to phase. Another model named RIMAM. This model takes care of the most frequently occurring risk and provides the way of handling all of them. This model is expected to cover all the handling and avoidance mechanism for software risks. In the series of the risk assessment models there is another model based approach named SRAEP which identify the risk by using the software fault tree (SFT). Risk identification, risk analysis and risk prioritization are the main subparts of model based approach. The above existing risk assessment models are quite different from each other in aspect of risk identification, risk mitigation, risk prioritization and risk avoidance. In this paper we proposed a new model (SRAIEM) software risk assessment identification and estimation model for the software risk management. This paper focus on take care of most frequently occurring risks and it is possible to predict the possible result of software projects with good accuracy proposed model not only access the risk but also identify and estimate the risk. This model provides a way to managing all of them. In order to cover some perspectives of the software industry which have not been covered up to now.

III. Software Risk Assessment models

There are few published models that evaluate the risk of software projects. Among the existing models some of them have been selected for detailed comparison in study. These risk strength assessment of the proposed models are expected to satisfy the needs of risk management for software development projects. In a survey only 35% of the projects succeed and, 49% are challenged and 16% fail completely. In our proposed model tried to measure the project failures, and concluded that only 38% project succeeds, 53% were challenged and 9% met complete failure. The model was selected because they are dedicated to manage the risks or related aspects. The models are described hereafter:

A: Software Risk Assessment and Estimation Model (SRAEM): In this model, the risk is estimated using risk exposure and software metrics of risk management and this metric is based on mission critical requirements stability risk metrics (MCRSRM). This model not only evaluate the risk but it also estimate the risk. Initially the model estimates the sources of uncertainty using different paradigms such as measurement error, model error and assumption error. The detailed description of this paradigm is given. This model also uses the concept of function points to explain these errors. This model is different from other existing models because the other models do not consider the issues related to requirement analysis.

B: Risk Identification, Mitigation and Avoidance Model for Handling Software Risk (RIMAM):

This model briefly presents the strategies that are expected for the purpose of identification, management and avoidance of certain risk factors such as Immature Requirements, Less Reusability, Delivery Deadline, Over-optimistic Technology Perceives, Staff Experience, Staff Turnover, Excessive Error Detection and Preservation of Intellectuals. The Detailed description of such risk factors. This model can be implemented easily with minimum cost. The easy and less costly processing may ensure that the already risk effected process may be completed more promptly and hence reducing the risk of further delay. This model can be customized with respect to the environment in which it is being used.

C: Software Risk Assessment and Evaluation Process using Model Based Approach (SRAEP):
This method explains a better technique of risk estimation and risk prioritization. In SRAEP, software fault tree approach (SFTA) is used to identify and analyse the risk. In SFTA, a high level threat is decomposed into intermediate objectives which can be further decomposed into individual attacker actions. Attacker actions are connected by AND or OR relationship. Generally, the exchange of an OR node with an AND node is expected to increase the safety of software development project. After the identification of risk, several countermeasures come into pictures for risks. To pick an appropriate countermeasure, a new technique known as Risk Reduction Leverage (RRL) which is a simple calculation that gives a numeric value to a countermeasure enabling different countermeasures to be compared. Mathematically it can be written as:

\[ \text{RRL} = \frac{\text{Reduction in Risk exposure}}{\text{Cost of Countermeasure}} \]

After computation of risk and value of RRL and its prioritization the next step is team review and action planning.

D: Software Risk Assessment Model (SRAM):

This model makes use of comprehensive Questionnaire. Test results show that using the risk indicator obtained from the SRAM, it is possible to predict the possible outcome of software projects with good accuracy. This model considers the following nine critical risk elements: complexity of software, staff involved in the project, targeted reliability, product requirements, method of estimation, method of monitoring, development process adopted, usability of software and tools used for development. A set of questions is carefully chosen for each of these elements with three choices of answers each. The answers are arranged in increasing order of risk. This model has considered the method of prioritization as a single step of risk assessment but do not specify how prioritization would be done.

E: Software Risk Assessment Identification and Estimation Model (SRAIEM):

In this model, the risk is estimated using risk exposure and software metrics of risk management and this metric is based on mission critical requirements stability risk metrics (MCRSRM) and presents the strategies that are expected for the purpose of identification, management and avoidance of certain risk factors such as Immature Requirements, Less Reusability, Delivery Deadline, Over-optimistic Technology Perceives, Staff Experience, Staff Turnover, Excessive Error Detection and Preservation of Intellectuals. This model not only evaluates the risk but it also estimates the risk and this model can be implemented easily with minimum cost. The easy and less costly processing may ensure that the already risk effected process may be completed more promptly and hence reducing the risk of further delay. This model can be customized with respect to the environment in which it is being used and also initially the model estimates the sources of uncertainty using different paradigms such as measurement error, model error and assumption error. In this model presents the strategies that are expected for the purpose of identification, management, avoidance of certain risks and estimated using risk exposure and software metrics of risk management.

IV. Strengths of our Proposed Model

- Our proposed model SRAIEM, models focus specifically on the Immature Requirements.
- Role of stakeholders and other software developers is fully cooperative in our proposed model.
- Our proposed model has predefined target reliability (i.e. error handling conditions, error tolerance conditions, hardware faults etc.).
- Our proposed model focus on evaluation of risk such as determines the level of risk, prioritize the risk and categorize the risk.
- Back-up issues have been seriously handled by almost all the risk assessment models by installing the repositories so that can one can move backward from a certain location.
- Various types of prediction methods have been used by our proposed model in order to estimate the sources of uncertainty measurement error and identify the risk.
V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a new Software Risk Assessment Identification and Estimation Model (SRAIEM) focuses specifically on the Immature Requirement and evaluation of risk such as determine the level of risk, prioritize the risk and categorize the risk. This model has predefined target reliability and various types of prediction methods have been used. In the proposed scheme Back-up issues have been seriously handled by almost all the risk assessment models by installing the repositories so that one can move backward from a certain location.

There is no specific model which is able to manage the risks in web distributed environment alone. For the effective risk management in above two areas (GDSD, Web Distributed), the software developer has to combine the strengths of various models and tackling the weaknesses of them in a new model is a step forward in improving risk management in these two areas.

REFERENCES