

Requirement Risk Identification: A Practitioner's Approach

Halima Sadia
Dept of Information Technology
Integral University
Lucknow, India
halima.sadia12@gmail.com

Md. Rizwan Beg
Dept of Computer Science
Integral University
Lucknow, India
rizwanbeg@gmail.com

Md. Faisal
Dept of Computer Application
Integral University
Lucknow, India
foreverfaisal@gmail.com

Abstract-Software project development has always been associated with high failure rate. The failure of a software project depends on rooted risks. These risks don't come into face at early stages of requirement gathering and analysis which carried on in the further phases of development greatly affects the cost of a software project. Researchers consider requirements uncertainty and risk as a problem to be addressed during the early stages of software development. It is a well known fact that it is more feasible to make changes to the software system under development in the early stages of the software development cycle. However, there are a very few techniques available for assessing risk at the requirements level and those that are available are highly subjective and are not based on any formal design models. In this paper, we are proposing to implement inspection technique to identify the key requirement risk factors responsible in achieving successful outcome and use a Bayesian network approach to establish a model for predicting the risks that leads to fail a software projects.

Keywords : Software development, Requirement Risk, Bayesian Analysis, Inspection Technique

I. INTRODUCTION

With the development of technology, software is becoming critical for the success of many large systems. When the development of a software project is carried out, the system's users, developers, and other stakeholders establish a set of requirements to be met by the system after development. A major risk factor affecting project schedule, budget and quality is the ability to successfully elicit requirements and execute on them. Standish Report [1] indicates that requirements, the 'contract' between customers, end users and the software development organization that defines what gets produced, "is a primary source of software project risk and software defects." Software requirement risk addresses the possibility of suffering a loss of any functional or non-functional requirement of the software system [2]. Requirement risk can lower the project performance therefore effective management of requirement risk is critical to achieve desired project performance. Identifying requirement risk at this stage usually is subjective and needs a panel of domain experts to analyze the requirements. Currently available techniques for assessing requirement-based risk are ad hoc in nature and are not based on any popular modeling techniques [2]. They are usually performed in an informal and

subjective way by a "risk team". But in reality, not all the software projects have the luxury of having a group of domain experts to carry out this analysis and such an assessment is completely dependent on the opinion of the risk assessment team [2].

Requirement Risks

Requirements are the foundation of the software development. They provide the basis for cost estimation and for planning project schedules as well as for designing and testing specifications [3]. It is more important to identify requirement risks at early stage. Additional research demonstrates that 82% of rework efforts is focused on correcting requirements defects; 13% is spent on fixing design errors, and contrary to popular belief, only 1% is spent on fixing coding errors [4,5]. The following requirement risk factors have been derived from literature and examinations of the previous systems [4], as appears below:

Requirement Risks	<ul style="list-style-type: none">• Poorly defined requirements• Ambiguous requirements• Volatile requirements• Inadequate requirements• Impossible requirements• Invalid requirements• Requirement not necessary• Requirement not verifiable• Requirement not attainable• Requirements not traceable to a parent
-------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

TABLE I VARIOUS REQUIREMENT RISKS

II. PROPOSED WORK

Our work is divided into two sections. The first section proposes a Risk Identification Process (RIP) based on inspection techniques for identification of requirement risk at an early stage. The second section proposes a method to prioritize the requirement risk using Bayesian Analysis depending on the relationship of these identified risks.

A. Risk Identification Process (RIP)

Inspection is a best known way of detecting defects in SRS [5]. We are using inspection technique to identify requirement risks. Although it is not an easy task to identify risk at an early stage yet many of the requirement inspection techniques like perspective based reading, ad hoc technique and checklist based reading can be introduced for detecting risk. In Requirement Identification Process five participants will play vital role through executing their individual responsibilities well to identify risks at early phase. These participants are (i) Moderator (ii) Author (iii) Reader (iv) Inspector (v) Recorder. After assigning the requirement inspection task to the participants; they are responsible to follow an appropriate method assigned to them for inspecting the requirement document [5]. Inspection process will have the following key steps:

Preparation: is a key stage in which members of the inspection team prepare individually for the inspection by reviewing and finding potential risks in the requirements.

Inspection Meeting: where team members as a group review the product to find, categorizes, and record risk requirements. Potential Risks found by individuals is discussed during the actual inspection meeting.

Third Hour: Optional additional time, apart from the inspection meeting, that can be used for discussing possible solutions.

Rework Stage: Required when there are a large number of risks. The moderator and the team decide the necessity for reinspection at the end of the inspection meeting.

Reinspection: Reinspection allows the changes to the product to be reviewed by the entire team instead of just the moderator.

Follow Up: Short meeting between the moderator and author. It determines whether risks found during the inspection meeting have been eliminated.

Our Risk Identification Process (RIP) has the following inputs:

- a) *Enterprise Environmental Factors:* Published information including commercial databases, academic or industry studies, benchmarking, studies, may also be useful in identifying risks.
- b) *Project Scope Statement:* Uncertainty in project assumptions should be evaluated potential causes of project risk.
- c) *Requirements:* Requirements are used as basic inputs in the RIP.

Output of Risk Identification Process (RIP): The output of the RIP is a risk log.

The Risk log contains list of identified risks, list of potential responses, root causes of risk and Updated risk categories which can be helpful for better management of risk.

Building the Requirement Risk Model

Risk is defined as a combination of two factors: probability of malfunctioning (failure) and the consequence of malfunctioning (severity) [2]. Once risks are identified, the identified risk must be analyzed to find their probability of occurrence and their relationship. The project team can determine how to address those risks which have the greatest potential probability and impact on the project. Bayesian analysis is a statistical method for supporting the decision making process by representing beliefs about the world as probabilities. These probabilities are not definitive, meaning reasonable people might disagree about the validity and applicability of the resulting data. However, given informed prior information, a reasonable and defensible inference about new data based on previous data can be made [6]. In order to apply the Bayesian approach, a method for combining information about a project's perceived level of risk with Bayesian reasoning must be established. This can be accomplished through the development and application of a Bayesian network. A Bayesian Network (BN), based on probability theory, is a knowledge representation that effectively captures the uncertainties and conditional independences present in a given Bayesian Network [7] (also called Bayesian Belief Network) is a cause-effect graph in which the nodes represent variables whose value is uncertain and the arcs represent dependency relationships between these variables. Bayesian Network becomes attractive due to its capability of modeling uncertainty effectively. It has been successfully deployed in certain domains such as medicine diagnosis [8] and software fields including software productivity prediction [9], project risk management [10], software defects prediction [11] and so on [12-16]. Bayesian parameter learning is to deduce the probability distribution of goal variables given the observed values of other variables. It is not reasonable to try to quantify every factor that could impact requirement risk for a system. This is what makes the Bayesian approach so valuable. It is also important to choose risk factors that are not domain specific, so the resulting BN can be used repeatedly and among many domains.

Risk Position	Risk Identification Time	Risk Indicator	Risk Cause	Risk Identification Cost	Risk Severity	Risk Priority	Risk Definition/Classification
Functional Requirement	00:00 hours of RIP	Symptoms	Potential Source of Risk	RIP Effort & time taken in identification	S_i	P_i	Category of Risk

The following are the characteristics that the Bayesian Network holds:

- Bayesian Network is mainly used in uncertain situation.
- Bayesian Network can combine information including historical project data and expert opinion. At the same time, domain knowledge might be used to identify whether critical data is missing and whether the collected data is suitable.
- The various factors leading to requirement risks needs to be transformed into knowledge and Bayesian Learning.
- Bayesian Network can be used for 'what-if analysis to explore the impact of changes in some nodes on the other nodes.'

According to Bayes' rule [6] we can compute the probability P (risk | positive) that the requirement given has a risk as:

$$P(\text{Risk} | \text{Positive}) = \frac{P(\text{Risk}) * P(\text{Positive} | \text{Risk})}{P(\text{Positive})}$$

Where P (Positive) is computed as the sum

$$P(\text{risk}) * P(\text{Positive} | \text{Risk}) + (1 - P(\text{Risk})) * (1 - P(\text{Negative} | \text{non risky requirements}))$$

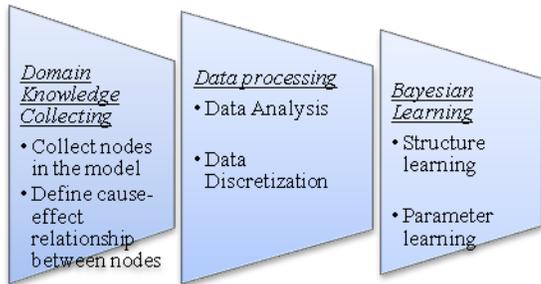


Fig.1. Flow of Bayesian Method

After analyzing the various characteristics of Bayesian Network, we finally choose it to prioritize Requirement Risks. In order to be a useful tool the Bayesian Network must be seeded with information describing the various requirement risk factors that have affected the systems in the past. Using the risk factors identified in the previous section as the nodes of the BN, a model has been constructed as follows:

III. CONCLUSION & FUTURE WORK

This paper outlines a method for predicting the level of requirement risk a system may experience during SDLC. The information used to develop the BN model for requirement risk has been taken from literature. In order to decide on the best actions that minimize or mitigate the effects of risk on software projects, the program manager must implement a method that represents his /her beliefs about each risk factor and make an inference about their impact to the overall software. Armed with this information, he can build risk mitigation planning into the program plan.

The next step of research is to collect & investigate the different requirement risk from industry experts. This collection of requirements can be performed by taking a web based survey and by interviewing the project managers taking their experiences regarding the requirement related risk. Any BN tool can be used to calculate the probabilities according to identified risks & their relationships.

REFERENCES

[1] Standish Report 2009
 [2] Software Requirement Risk Assessment Using UML.
 [3] K. E. Wiegers, "Software requirements", Microsoft press 1999.
 [4] Lamswede, A. Requirements Engineering in the Year 00: A research perspective. In proceeding of the 22nd International conference on Software Engineering (ICSE'2000), Limerick, Ireland, 5-19, ACM Press
 [5] S. Nayak, R. Khan and R. Beg "Evaluation of Requirement Defects: An Implementation of Identification Technique" ICIET 2012, Journal of Procedia Technology
 [6] Laskey, K., *Bayesian Inference and Decision Theory*, SYST 644 Class Notes .Dept. of Systems Engineering, George Mason University, <http://ite.gmu.edu/~klaskey/SYST664/SYST664.html> Spring 2003.
 [7] Lan Yuqing, Zhao Tong, Bayesian Network to Construct Interoperability Model of Open Source Software, International

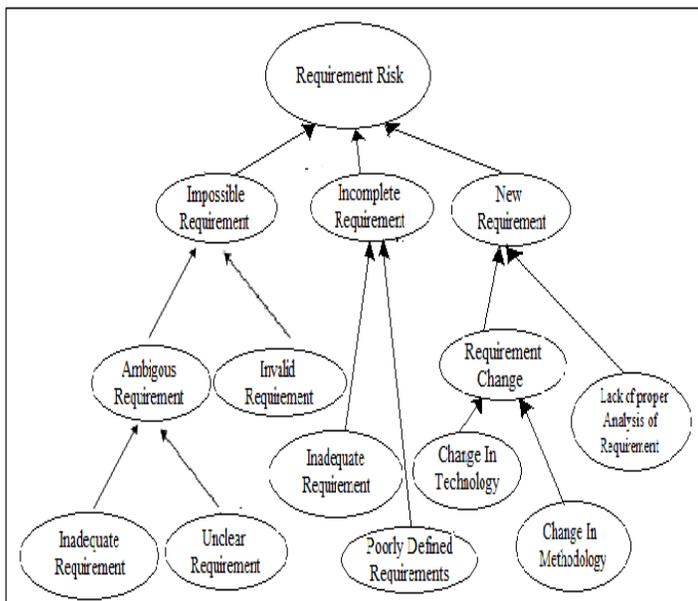


Fig.2. The Bayesian Structure for Requirement Risk

Conference on Computer Science and Software Engineering, 2008

[8] Deborah Ashby, Bayesian statistics in medicine, Paper Celebrating the 25th Anniversary of Statistics in Medicine, STATISTICS IN MEDICINE, 2006.

[9] Stamelos, I., Angelis, L. P., & Sakellaris, E. on the use of Bayesian Belief Network for the prediction of software productivity, Information and Software Technology, 2003.

[10] Ward, S., & Chapman, C. Transforming project risk management into project uncertainty management, International Journal of Project Management, 2003.

[11] Norman Fenton, Martin Neil, et al, Predicting software defects in varying development lifecycles using Bayesian nets. Information and Software Technology 49, 2007.

[12] Wooff, A. D., Goldstein, M., & Coolen, P. A. F., Bayesian

graphical models for software testing, IEEE Transactions on Software Engineering, 2002.

[13] Y. P. Wu, Q. P. Hu, K. L. Poh, S. H. Ng and M. Xie, Bayesian Networks Modeling for Software Inspection Effectiveness, 11th Pacific Rim International Symposium on Dependable Computing, 2005

[14] Cooper G. Computational complexity of probabilistic inference using Bayesian belief networks [A]. Artificial Intelligence. 1990

[15] Neil M., Littlewood B., Fenton N., Applying Bayesian Belief Networks to Systems Dependability Assessment. Proceedings of 4th Safety Critical Systems Symposium, Germany: Springer Verlag, 1996

[16] Xiaoxu Wang "Software Project Schedule Variance Prediction Using Bayesian Network" 2010 IEEE