

A Model For The Proactive Risk Management Based On The Text Mining Classification

M.S. Rojabanu¹, Dr. K. Alagarsamy²

¹Research Scholar, Madurai Kamaraj University, Madurai, India .

²Associate Professor, Computer Centre, Madurai Kamaraj University, Madurai.

Abstract

Identification and controlling the software risks, enables one to make better and more daring decisions when taking on complex challenging projects or when exploring new unknown grounds. This paper proposes a new model for the proactive risk management based on the Text Mining classification. The Model is discussed, the possibilities of building such model and the outcome is also discussed.

Keywords: Proactive risk Management, Text Mining, Classification.

1. Introduction

Software Risk management is a challenging job in the area of software project management. The software risk can be tackled by the sequence of activities which include the identification of the risk, the identified risk is to be analyzed and it is followed by the risk evaluation. If the risk evaluated is beyond the acceptance level then the risk mitigation starts with the following the controlling measures and after controlling the learning of the risk takes place. It is observed that for the project to be successful, the risk management must be run as a continuous process involving repeated risk assessment and project-wide risk mitigation.

Risk is a function of the likelihood of a given threat-source's exercising a particular potential vulnerability, and the resulting impact of that adverse event on the organization [1]. Risk can be defined as the probability of an unpleasant event occurring and its impact. The impact manifests itself in a combination of financial loss, time delay, and loss of performance [2]. The

extend of the risk impact determines the importance of the risk management. In a time of ever-increasing quality demands and ever-shortening time-to-market windows, software development is a riskier business proposition than ever [3].

A proactive software risk management approach can be implemented through a collection of risks and by the decision making method. Successful software projects could be developed by dealing with risks by recognizing and minimizing uncertainty and by proactively addressing each identified risk. Proactive risk management means that the collection of risks or a risk database has a clear measurable method for managing risks. The proposed approach to risk management describes an environment in which the risk management system proactively analyses on a continuous basis, what can go wrong and then makes proactive choices about which risks need to be managed and manages them.

2. Project Risk Management

Risk management in software projects has different uses. It helps to save projects from failing due to different factors such as non-completion of projects within the specified schedule, and budget constraints, and not meeting customer expectations. Software project development has always been associated with high failure rate [4]. Risk management is the set of activities used to manage risks. Risk management usually consists of four basic processes: Risk Identification, Risk Analysis, Risk Planning/Mitigation, and finally Risk Monitoring and Controlling [5]. Risk assessment methods are one of the most important elements in the process of risk management. They are a vital element of software project management. These methods consider numerous aspects while assessing and estimating the risks.

2.1 Risk Identification

Early awareness of possible problems forms the basis of successful risk mitigation. Thus the risk identification is always the first phase of the risk management process. Once identified, the risk can be communicated within the project and then analyzed and coped with by undertaking appropriate actions [6].

2.2 Risk Analysis

In software engineering, risk analysis is used to identify the high-risk elements of a project. It provides ways of documenting the impact of risk mitigation strategies. Risk analysis has also been shown to be important in the software design phase to evaluate criticality of the system, where risks are analyzed and necessary countermeasures are introduced [7]. The purpose of risk analysis is to understand risk better and to verify and correct attributes. A successful analysis includes essential elements like problem definition, problem formulation, data collection [8].

2.3 Risk Planning/Mitigation

The second process of risk management, involves prioritizing, evaluating, and implementing the appropriate risk-reducing controls recommended from the risk assessment process. Because the elimination of all risk is usually impractical or close to impossible, it is the responsibility of senior management and functional and business managers to use the least-cost approach and implement the most appropriate controls to decrease mission risk to an acceptable level, with minimal adverse impact on the organization's resources and mission [9].

2.4 Risk Monitoring and Controlling

In this part of the risk management, continuously monitoring and controlling of the risks according to the risk management plan takes place. It can be also used for identification of new risks. Risk monitoring procedures must be created for the effective monitoring and control. For each risk or risk group, continuous monitoring and records the status are kept [10]. In cases when the status changes, one takes measures as specified in the plan. Finally, updating and recording of the risk status is done [11].

3. Proactive risk management

Before applying any risk management process, the project team members should be clear about the following dimensions of risks in their projects like the nature of uncertainty involved, and the likelihood with which the risk will occur, the loss that will be incurred if the risk occurs. Loss in software projects can take many forms including loss of revenue, loss of market share, and loss of customer goodwill. The severity of the loss and the duration of the risks are also to be considered [12].

In this paper we have proposed an approach based on the data mining techniques applied for the proactive risk management. The out line of the proposed methodology is given by the following figure 1. The model

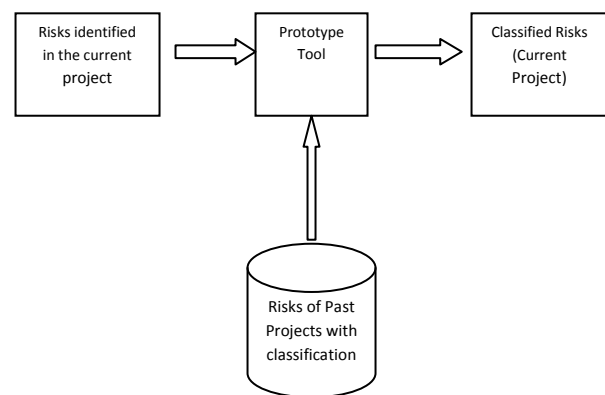


Figure 1: Proposed Method for the Proactive Risk Management

The Approach starts with the risk analysis which is discussed in section 2.2. The risk analysis is followed by the building of the risk repository. The repository of the risk could be built by the following methods

1. By interviewing
2. By the historical review of previous like projects
3. By the experts report
4. Analysis of high-level deliverables
5. Analysis of the WBS and project schedule
6. Analysis of scope change requests
7. Analysis of project assumptions
8. Project team input (which can take the form of interviews, brainstorming sessions, and/or Delphi technique)

Risk Management Matrix (Risk Register)													
Project						Project #							
Project manager						Sponsor			0				
Project artifacts						Updated							
ID	Risk Description	Probability	Impact	Detectability	Importance	Category	Trigger Event/Indicator	Risk Response and Description	Contingency Plan	Owner	Status	Date Entered	Date to Review
1	What is this risk?						What act or event initiates either the risk occurrence or precipitates the response strategy?	How will you respond to this risk and what actions will you take to match that response?	If the risk becomes a reality, what will you do in response, as a backup, or alternative/ workaround?	Who monitors this risk?			

Figure 2: Sample risk register (Source: www.iappm.org)

9. Stakeholder and sponsor input
10. Formal risk identification sessions
11. Previous lessons learned
11. SQA audits and reviews
12. Performance and status reports
13. Diagramming techniques such as cause and effect diagrams, process or system flows, and influence diagrams.

Once the repository is built the process is ready for the proactive risk management. The Proactive risk management starts with the selection of the risk category to be identified. The selected risks are then applied for finding the frequent risk occurrences. The next step in the proposed model is applying text mining.

The text mining phase is followed by the Classification of the risks, the classification enables the risks to be categorized and then the proactive management of the risk can be done. The pro action could be enabled by the expected risk based on the classified risks. The risks could be concentrated and the risk management could be done effectively.

3.1 Methodology

A Software project undertaken by a local software organisation was the subject of study. A database of risks of past projects undertaken by the organisation is available. The CART algorithm for Text Mining was implemented for identifying the risks

in the current project that are very similar to the ones in the past project. For all the risks of the past project, the classification level with respect to the impact, probability of occurrence is available.

The implemented prototype tool uses the CART algorithm for text mining for identifying the similarities of the risks in the current project with that in the previous ones and using the classification level of the previous ones, classifies the risks of the current project.

3.2 Experiment details

The experiment is carried out from the construction of the risk register. The model template of the risk register is given as below. The risk register consists of the following details such as risk description, probability of the risk, impact of the risk and such details.

The risk is then added in the risk archive and the repository is built, the selected like risk are then given for the frequent risk mining and the rules are constructed. The Rules are then classified by the precision and accuracy. The classified risks are provided for the documentation of the proactive risk management based on the

- Risk impact
- Risk probability

- Risk matrix score – computed by the risk register spreadsheet after impact and probability are entered
- Risk priority – computed by the risk register spreadsheet after impact and probability are entered
- Qualitative impact – descriptive comments about the potential risk impact

3.3 Results : Types of Risks [13]

Rank	Software Risk
R1	Unrealistic time and cost estimates
R2	Developing the wrong software functions
R3	Developing the wrong user interface
R4	Gold Planning
R5	Late Changes to requirements
R6	Shortfalls of externally performed tasks
R7	Shortfalls of external supplied components
R8	Real-time performance shortfalls
R9	Straining science capabilities
R10	Lack of top management commitment to the project
R11	Misunderstanding of the requirements
R12	Not managing change properly
R13	Failure to gain user commitment
R14	Lack of effective project management skill
R15	Lack of adequate user involvement

Table 1: Risks in each category – impact

High impact risks	Medium impact risks	Low impact risks
R1,R2,R3,R5	R4,R10,R11,R12	R6,R7,R8,R9, R13,R14,R15

Table 2: Risks in each category – Probability

High Probability risks	Medium Probability risks	Low Probability risks
R6,R9,R11,R12, R13,R15	R1,R2,R3,R5,R, R10,R14	R4,R8

Table 3: Percentage of Risks in each category - Impact

High impact risks	20%
Medium impact risks	33%
Low impact risks	47%

Table 4: Percentage of Risks in each category - Probability

High Probability risks	32%
Medium Probability risks	35%
Low Probability risks	13%

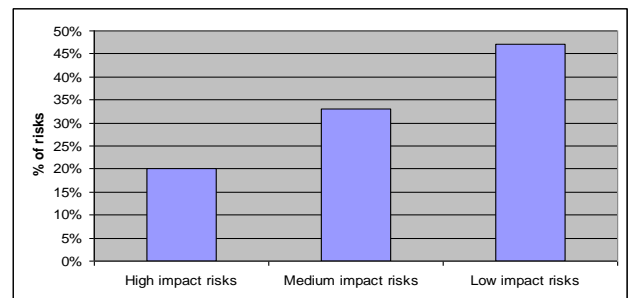


Figure 1: Percentage of risks in each category - impact

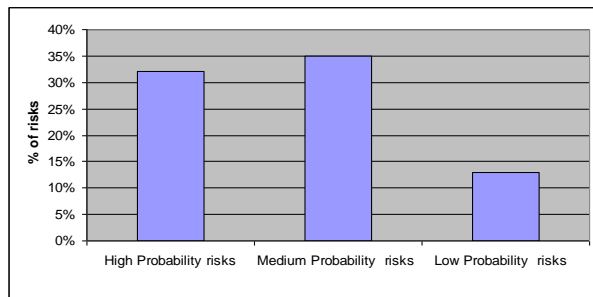


Figure 1: Percentage of risks in each category - Probability

3.3 Discussion

Using the implemented tool, the project manager can make very important decisions. For example if he finds an increased percentage of high impact risks, he should strive to minimize it. Given insufficient resources for mitigating all risks, he can make crucial decisions pertaining to the usage of resources for the high impact risks and high probability risks. Therefore the developed tool will be extremely valuable for the project manager. As can be observed the percentage of low impact risks is higher than medium and high impact risks and the percentage of medium probability risks outnumbers low and high probability risks.

4. Conclusion

Text mining can be of vital help in risk management. It can be effectively used for categorizing the risks in the current project. As the organization develops and releases more software products, the risk database will contain more accurate information and the outcome of the implemented tool will become extremely reliable. The tool will be of great help to software project manager in risk management and to reduce loss.

References:

- [1] NIST Risk Management Guide for Information Systems Special Publication 800-30. July, 2002
- [2] Abdullah Al Murad Chowdhury, Shamsul Arefeen, Software Risk Management: Importance and Practices, IJCIT, VOLUME 02, ISSUE 01, July 2011.
- [3] Will, Brian A. Software Risk Management. White Paper. Encinitas, CA: Paroxys, LLC, 20004.

[4] Hu Yong , Chen Juhua , Rong Zhenbang , Mei Liu , Xie Kang, A Neural Networks Approach for Software Risk Analysis, Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops, p.722-725, December 18-22, 2006

[5] Ian Sommerville, "Software Engineering", Addison Wesley, 7th Edition, 2007.

[6] Jakub MILER, Janusz GORSKI, "Risk Identification Patterns For Software Projects", Foundations of Computing and Decision Sciences Vol. 29, No. 1-2, 2004, pp. 115-131

[7] Yudistira Asnar, Paolo Giorgini, "Risk Analysis as part of the Requirements Engineering Process" University of Trento, Department of Information and Communication Technology, 2007

[8] Bryan L. McKinney, David R. Engfer, "Formulating Risk into Research and Engineering Projects", Crystal Ball User Conference, 2004

[9] Abdullah Al Murad Chowdhury and Shamsul Arefeen, "Software Risk Management: Importance and Practices", IJCIT, VOLUME 02, ISSUE 01.

[10] Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK), 3rd Ed. ANSI/PMI 99-001-2004, PMI, Newton Square, PA, 2004.

[11] IEEE 1540, IEEE 1540 Standard for Lifecycle Processes-Risk Management. IEEE, New York, NY, 2001.

[12] Smith, P. and R. Pichler (2005). Agile Risks/Agile Rewards. Software Development, 13(4), 50-53

[13] Tharwon Arnuphaptrairong, Top Ten Lists of Software Projects Risks: Evidence from the Literature Survey, IMECS 2011, Vol.I, March 2011.