



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijircce.com](http://www.ijircce.com)

Vol. 5, Issue 4, April 2017

## A Survey on Software Risk Management Methods

Shubhendu Shekhar<sup>1</sup>, Raghav Srivastava<sup>2</sup>

B.Tech Student, Computer Science, VIT University, Vellore, Tamil Nadu, India<sup>1</sup>

B.Tech Student, Computer Science, VIT University, Vellore, Tamil Nadu, India<sup>2</sup>

**ABSTRACT:** Software risks are generally the financial or resource losses, the occurrence of potential problem that may occur in any phase of the software development in an organisation. Software risks can be viewed as a collection of performance efficiency, robustness, transactional risks and security occurring in a system. These may be caused due to lack of information at a specific level of development or lack of time and control which results in a catastrophic failure of the project. In this review paper, some of the software risk management methods have been discussed which have been successful in their work for reducing the risks associated with the software projects. Software risk management still is not being used by many of the firms because of the lack of research in that field. It is a prior requirement in today's generation of technology oriented world to create software faster, efficiently and with enough features as required by the people.

**KEYWORDS:** Risk management; projects; development; organisations; efficient.

### I. INTRODUCTION

In today's world, the technology is taking over almost all aspects of life, it maybe manufacturing, healthcare, simulations, etc. The manual labour is being replaced by the automated machines and to run these machines certain software are developed which controls all the functioning of the machines. Software development had been going on for quite a long time and there have been many advancements in the field of software development both in the field of production and research to create better software for everyone. But, there are some of the risks associated with software project development which maybe the losses incurred by the company due to lack of time, information, resources. Sometimes the software development projects fail miserably and would end up being finished late after the deadline or may not have the required features as demanded by the users. Therefore, the software risk management methods have been introduced which find the risks associated with a project, analyse it and then take proper steps to reduce those risks. A number of different approaches for risk management has been discussed in this paper. These software risk management techniques would eventually help to reduce the risks associated with software development and would help to produce software n time and with increased efficiency and features.

### II. LITERATURE REVIEW

Software risk management is vital in every successful project. Every phase of software development life cycle(SDLC) remains significant sources of risks as it involves software, hardware, people, technology, schedule and cost. In paper [1], we use a software risk estimation model which is probabilistic in nature called Bayesian Belief Network(BBN) which focuses on high priority software risk indicators for the assessment of risks in projects. In current conditions, the impacting factor for failure of software and large amount of low quality software. Often the technologies to understand risk management is neglected which in turn produces faulty software. The BBN is a technique which models the relationships of the datasets or systems which are casual in nature and are represented through use of directed acyclic graphs(DAG). This network is based on Bayes Theorem which is expressed as:

$$P(R/S) = \frac{P(S/R)P(R)}{P(S)}$$



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijircce.com](http://www.ijircce.com)

Vol. 5, Issue 4, April 2017

The probability distribution which is related to each node or the quantitative part of BBN has a set of possible values called its states namely: 'good' and 'bad' or 'high' and 'low'. Therefore, decision on each node is made by supplying the evidence based on the given technique. The used methodology has the following steps:

- i). Selecting the top ranked indicator of software risk in the development project.
- ii). Construction of casual relationships between the metrics of risk indicators.
- iii). Constructing node probability table for every node.
- iv). Calculation of probability value of risk of software.

This model easily estimates the probability value of the software risk with the help of qualitative value of the indicators of software risks. The model is different from the current models as the present models do not consider the casual relationships. Tests have shown that this model has a much better accuracy of estimation, hence this model is used.

Paper [2], Innovative Software Project contains high levels of complexity and uncertainty which leads us to manage the risks associated with these projects. Using the innovation management would be a major factor in the success of the project. The innovative model maybe unpredictable to high levels but the uncertainty levels would vary according to the different factors. According to the authors of paper [2], the innovative models should have the following criteria: (i). development of innovative service or product (ii). applying methods and approaches which are innovative (iii). improving the capacity of organizational learning and innovation (iv). having a high level of integration with the owner of the project. Systematic Literature Review(SLR) has ways to make the results have a scientific value without biased literature reviews. It has 3 main parts: planning, conducting and reporting of review. This helps to identify the factors which foster innovation and minimize the effects of uncertainty in innovative projects and how the project managers and the teams would prepare for challenges in scenarios of the project based on SLR. The innovation projects are unstructured, the future is not certain also the conventional tools for project management are not useful in our context. Use and creation of collective intelligence has been applied to the fields like distributed mobile and robotics and pervasive systems. The findings in paper [2] shows that innovation projects have been growing in the last decade as well as increased awareness of the risks for the project management is identified. These findings contribute to software project management which provides better understanding of the risks dealing with ISPM, which can support researchers to identify challenges and to develop solutions for the projects.

In paper [3], multiple project management is discussed which is a reality in the environments of software development. Many aspects of software development like constant changes in scope of product, aspects related to human resources and the complexity of the software are considered as risks factors that have a dire need to be managed. A metric based strategy is used in [3] and a metric called "Risk Point" is used for identification of points of adjustments. Risks in a project are those events that could adversely affect the organisation environment and the development of projects. Risk management mainly comprises of identifying, analysing, planning and controlling the events that threaten the environment or reduce damage of these events in the case of occurrence. Use of metrics in risk management would make it more efficient and can minimize the degree of subjectivity which would facilitate understanding of risks and diffuse the application in environment of software development. The Risk Point Metric was developed to represent the complete exposure levels of risks for a project, is defined as:

$$RE(\text{Risk}) = \text{Probability}(\text{Risk}) \times \text{Impact}(\text{Risk})$$

Risk Point also allows quantification of project in terms of identified risks, necessary for estimating Risk Exposure value:

$$RP = PCF \times URPW$$

Here PCF is Project Characteristics Factor and URPW is Unadjusted Risk Point Weight. The URPW value represents overall evaluation of the known project risks which in turn allows wide assessment of risks and ways to compare different projects based on their risks. It is possible to assess the trends in growth and decrease in levels of risk exposure of each project, compare projects and evaluate efficiency of resources and actions applied in the life cycle of



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

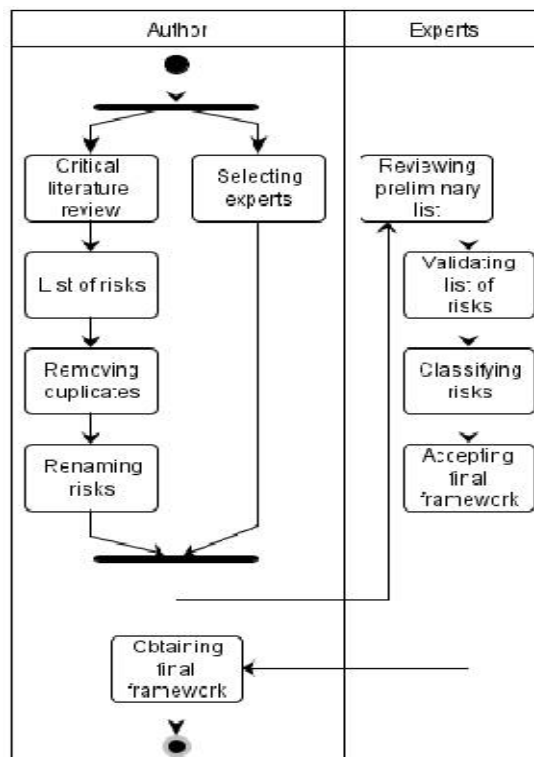
Website: [www.ijircce.com](http://www.ijircce.com)

Vol. 5, Issue 4, April 2017

the project, once all the metrics have been calculated. This idea was aimed to provide improve metrics sensitivity providing alternatives, quantifying risks providing tools of support for decision.

Paper [4] discusses the various types of risks that maybe connected to property, liability of 3rd parties, decisions or staff and also the various software tools for risk identification and analysis are discussed. Modernisation of workplaces has brought computers at every organisation and the IT dependency levels have come up which indicates that there is an equal chance of failure of these systems. Risks maybe defined as a synergy of probability of event and the consequences of it, for systems it is needful to recognize the different risks and analyse them. The different types of risks mentioned here are - Strategic risk, Business/financial risk, Program and project risk, Operational risk and Technological risk. These risks maybe classified into these risk classes - Technical risk, External risk, Environmental risk, Organizational risk, Project Management risk, Right of way risk, Construction risk and Regulatory risk. RISK tool is an add-in for MS-Excel which takes model of the spreadsheet and then ca represents a wide range of possible outcomes. This tool can be used for software development fields as the software projects can go over budget, cross deadlines and can cause loss of money, this toolbar can help the users to perform simulations and come up with possible risks associated with the different projects. The classification of risks based on the criteria provided ease in decision making for the tool, but the challenge for future use of this tool is the tedious amounts of manual entry of data into the system to perform risk estimations. In future, inclusion of fuzzy methods and artificial intelligence would further increase efficiency and improve the decision making for these tools.

The software maintenance projects contain features different from other ones, which include increase in complexity and more failure rates of projects. Hence to increase the chances of these projects to be a success, it's mandatory to identify and monitor the risks. In paper [5] a framework based on IEEE 1074 is discussed which would help in controlling the risk factors in software projects. Software maintenance is one of the critical stages in the life cycle, these are developed for removing bugs, enhancing performance and other factors and adjusting them in a dynamic environment. Software maintenance is recognised in the following categories: Corrective maintenance, Adaptive maintenance, Perfective maintenance, Preventive maintenance, User support. Failure in software maintenance may cause high profile disasters and would impact the stability of the company by hindering the daily business activities. Despite the consequences, very little effort is made to monitor the risks associated with software development. So, we use this mentioned framework which would find and analyse the risks and provide the methods to tackle those risks in the race for a successful software project.



# International Journal of Innovative Research in Computer and Communication Engineering

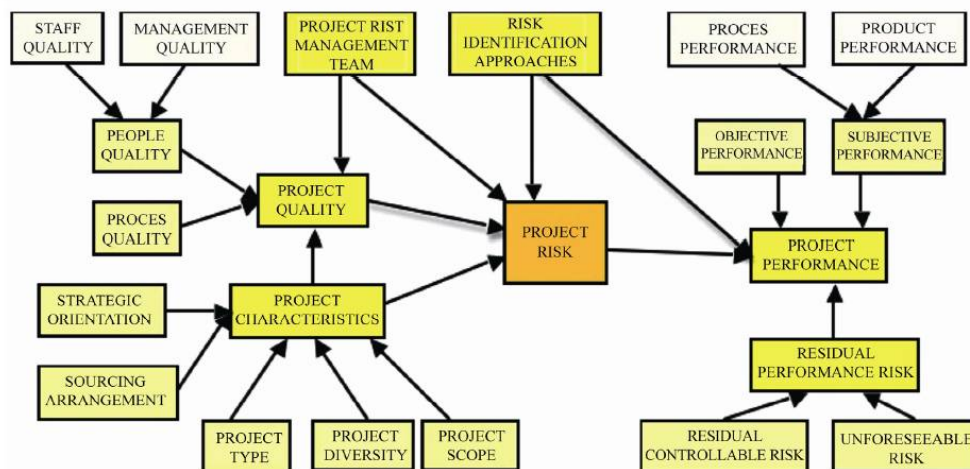
(An ISO 3297: 2007 Certified Organization)

Website: [www.ijircce.com](http://www.ijircce.com)

Vol. 5, Issue 4, April 2017

The final outcome of this framework contains the classified risks in the phase of maintenance that they influence. Moreover, risks of software project are focused on the development phase as it is one of the early stages in which modifications are possible, but many fail to work by this idea and end up in the failure of the project. If a proper risk management software is used for the software development projects it would decrease the failure rates, but still there are a number of methodologies to work on to use the tool on actual projects.

The adoption of risk management strategies has impacted the projects overall outcome and the organisations have realised its role in reduction of costs, improving performance and decrease in schedule overruns. In [6] a range of relevant literature is reviewed, new conceptual factors are introduced, a conceptual framework is introduced and new prospects in the risk management of software development is discussed. It is a fact that many organisations have a different perception of risk which increase the ineffectiveness and instability in the development of project. The increasing complexity of software projects and the risks associated in the development phase, human intuition maybe effective but is not always a sufficient approach to manage the risks. There are certain factors which characterize the stages of a project and may also be a determination factor of risk levels, the factors are: Project scope, whether project is in house or a collaborated one, Strategic Orientation, Project diversity and Type of system which is being developed. Conducted studies found 3 main dimensions of risk namely project structure, project size and the technological experience.



Specifically, project risk management in the field of software development is well grounded, small and a distinct system which would allow project managers to use a successful and reliable mechanism of management. The model discussed in paper [6] is yet to be tested with real life data, and then it would be a reality to use this framework in the risk management methods.

Review paper [7] discusses the positive influence that is gained by controlling of risks in software projects. For supporting of analysis the risk management status by the Australian government has been considered, which has shown that risk management has been practically under-performed. Some of the benefits of risk management are -

- favourable alternative actions are identified
- confidence in striving objectives is increased
- chance of success is improved
- surprises are reduced
- estimates are given with more precision
- effort duplication is reduced

There are certain risk response strategies that are found in this literature are - Avoidance, Transference, Mitigation and Acceptance. On overall these strategies are proven effective in supplying options for considerations to formulate



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijircce.com](http://www.ijircce.com)

Vol. 5, Issue 4, April 2017

responses of expected threats related to the projects. In the end the research of [7] concluded with three results i.e. risk management has a prior role in projects, development and conceptualization of both risks and risk management theory lags practicing to handle the threats associated with complete spectrum of uncertainties confronted by software projects and lastly, risk management practice lags the understandings and prescriptions of the risk management. By seeing the potential cost and loss from failed projects, practitioners and researchers must continue to learn to reduce failures of project and develop better outcomes as organizational capability is critical to achieve those objectives.

In [8], the states of risk management and risks are investigated in process models for software development. Actually, it capacities directors think up the philosophy that of the primary water suits their activities. So as to make extend progress, we trust that the most ideal approach to oversee hazards in programming activities is to choose the most reasonable strategy that best fits the proposed extend, and to consider it amid the advancement procedure as an intend to oversee dangers. Distinctive programming advancement systems bolster hazard administration by nature in variation levels. Further in this paper, the various software development models like waterfall model, V-model, incremental model, spiral model and agile models are discussed and also their major drawbacks faced in the course of work are highlighted, which creates a need for strong risk management techniques in the field of software project development. Thus, the paper [8] found that some product improvement strategies characteristically include hazard administration. For every approach, this requires certain conditions to exist. This shows dangers are inescapable in most programming advancement systems, and that all product advancement approaches, including the chance driven ones, require that hazard administration be upgraded in it.

Paper [9] discussed is generally a review paper on the techniques of Software Risk Management. For a successful managing of software projects and reap the rewards, it must be notes that it is important to identify, analyse and control the risks. The various types of risks that maybe encountered are –

- Technical risks
- Management risks
- Financial risks
- Contractual and legal risks
- Personnel risks
- Other resource risks

A plan of risk management is created which identifies the actions of containment and also reduces the probability of occurring of risks or reduces the impact of the risk. Risk identification involves a number of techniques like decomposition, interviewing/brainstorming, voluntary reporting, critical path analysis, assumption analysis and utilizing the taxonomies of risks. During the risk analysis, each of the risks are assessed to determine its likelihood, impact and timeframe. There is a risk equation which helps establish the risk priorities quantitatively, RE measures the risk impact in terms of expected value of loss.

$$RE = \text{Probability(UO)} * \text{Loss(UO)}$$

where UO = Unexpected Outcome

From the further discussions in the [9] it is concluded that risk management is an everlasting process which is implemented as initial project planning activities. It requires an environment which is fear-free where risks can be discussed openly. By adopting a proactive approach, it can even eliminate the crisis occurring in the software projects.

In [10] the software risk management approach is listed out with the various actions that have to be taken with it namely: risk list, risk action list, risk strategy analysis and risk strategy model. The paper [10] had tried to answer questions like, “What are the approaches frequently used for risk management?”, “How to implement risk management?”, etc. Risk management strategies have been used in many areas like nuclear reactors, construction, space exploration, etc. And now the time has come to use it in the software projects to create projects with minimum losses and in a comparatively less time. The various traditional approaches used in other areas which might also be useful in the software project development are –

1. Ad-hoc approach: Provides risk valuation when initial symptoms appear on the project.
2. Informal approach: Includes discussion with people, who maybe indirectly or directly involved relating to risk issues and documenting and recording the risks.



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijircce.com](http://www.ijircce.com)

Vol. 5, Issue 4, April 2017

3. Periodic approach: Repetitive procedures are involved, for specifying and identifying risks.

4. Formal approach: Detailed and in-depth assessment of risks are performed by individuals.

As discussed in the previous paper [9], the concept of risk exposure is given in this paper also, making use of probability of unexpected outcomes. The reports of 2004 indicate that 53% software projects have been denied to deliver within budget and on schedule, this could have been avoided if proper risk evasion strategies would have been employed at the correct time. It had been proven that software risk management is a sequential activity and emphasizes the communication between the stakeholders, and due to this sequential nature each stage needs proper risk management.

In the research paper [11] first a formula to compute the expected risk in any particular software development project is devised as a combination of the events, their probabilities and consequences. It suggests avoiding the goals that are not as rewarding when compared to the risks they carry. Another method is evaluation of the risk related to a particular event and gathering information regarding the various methodologies that can be used to minimize them. The various types of risk are technical, cost related, management, legal and personnel risks. Project Schedule also carries certain uncertainty with it, adding to risk. The two major phases involved with risk management are assessment and control. In assessment analysis and prioritisation of risk are major phases, in control mitigation and monitoring is involved. In risk analysis the probability, effect and the timeframe within which it can be rectified. The additional efforts that will be made to reduce risks has to be added in the project timeline. Depending on risk involved and importance of the task other tasks may be shifted. The results that are obtained on application of risk mitigation techniques should be tracked properly and the obtained data is to be analysed properly to improve upon the risk analysis methodology. The analysis helps in discovering new risks and validating that a particular risk has been avoided successfully.

In the research paper [12], choosing of correct methodology for managing risk is considered as the most vital step. The method should be according to the project and should also change according to the changes that occur throughout the Software Development Life Cycle (SDLC). The various software development models are waterfall, V-model, incremental, spiral and agile. In waterfall model the development occurs in a sequential manner, which means that an error, once made, will cascade down to the next phases of software development. At each step a review of the results from previous stage can be done, but this process becomes iterative and extremely complex. A different method in which a design phase is added before analysis so as to limit the analysts, this is achieved by constant iteration between analysis and design so as to come up with an acceptable design phase. Still waterfall remains risky as it won't be able to handle constant changes in the requirements that happen at each phase. An incremental model is proposed in which the project is broken down into various segments. This also breaks down risks and makes them more manageable. After reducing risk of each segment, the segments are combined together. Mini additions to the project can be done later after receiving feedback from the user. Each addition is analysed for risks before adding it to the software. However, each addition comes with its share of requirements. Each subsequent increment makes it harder to address an error that might have occurred at an initial stage. As an improvement V-model is used in which testing and development are given equal importance. After every phase testing is done. V+ model is a modification of V-model which adds user involvement as z-axis. Spiral model is a risk driven model. We identify the most threatening risks and deal with them first. It is highly dependent on human expertise as risk identification is the first step of the model and is less cost effective. In Agile model the development starts without waiting for software requirement specification. In this constant small increments keep on happening based on customer review. At each step user requirement is gathered with constant user feedback. This method is ineffective for large projects as increments will be more and communication between developers will be hard.

In the paper [13] risk management techniques for websites and distributed projects. Each software can be divided into three categories: product, process, and project. A risk can affect only a process or a product or the entire project itself. For instance, an incompetent software developer will risk the entire project whereas poor testing techniques will affect only a particular process. In a distributed software development project different components of the project are developed by different teams located at different places. Various challenges like miscommunication, knowledge difference, cultural difference, and time-zone difference can lead to risks in the project. In web based projects security is a major risk factor and as the project is deployed worldwide instantly more number of people will be affected with any sort of vulnerability that is present in the website. The risks involved in distributed software projects are also there in



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijirccce.com](http://www.ijirccce.com)

Vol. 5, Issue 4, April 2017

web based project as often various parts of website are made by different developers or code already implemented by a different developer is reused to design the project. Majority of web based projects are able to detect vulnerability only if something goes wrong with the website, which makes such projects hard to manage. Various methodologies are discussed which can be used to manage risks in web based projects. The EBIOS methodology focuses on Information Security risk, it provides various guidelines to make a web based project more secure. In ProRisk the developer has pre-designed templates available with him and tries to fit his project in any one of them. In SoftRisk all the steps of development are documented carefully, and based on this documentation and analysis of past data available the developer comes to know about the most important risks which need to be addressed first. In PMBOK process the software development is divided into four phases namely- Initiation, Planning, Execution and Termination. Various types of knowledge are used in this model to carefully assess the four phases for risks. In Performance and Risk Model a link between risk and performance analysis is made. The importance of each risk is evaluated on the basis of change in performance efficiency of the project if it is left as is. An analysis is done on each model and it is deduced that no particular model fits all web based projects, the factors that make one model more suitable can make another model unsuitable.

In the paper [14] usability testing is incorporated. In the risk definition process we define risks that are unacceptable so that in the phase of testing they can be identified out easily. In contemporary usability testing all expected hazards are listed and tested for. The paper tries to investigate what varieties of risks can be recognized using usability testing. A scenario of a hospital is considered and for usability testing only 4 controls are used. Various risk analysis meetings were held during the course of experiment in which detectability of each hazard was assessed. The risk coefficient is calculated as product of probability of occurrence of that risk and severity of that risk. The data received from usability test on the 4 controls was also recorded and divided into any one of the three sections: alarm, comments and others. The results of usability tests were combined with that of risk analysis, then the usability were divided into two categories based on whether they are related or unrelated to any identified risk. It was found that only 40% of usability issues were originally identified as risks. The results of the usability tests helped recalculate the value of risk coefficient, as the probability of occurrence of each risk was changed based on the result, also the severity of risk was also changed depending on whether the patients found the issue discerning. Only the risks that were there in comments section were added to the previously kept list of risks. The usability testing also helps the developer get feedback on how the user accepts new functionality that has been introduced. It also tells whether user is unaware of a particular functionality and how the user approaches the product compared to how developers wants him to. It can also be used to verify that if a risk is given high value during the development phase then whether it actually is worthy of its value or user will just ignore that risk.

In the paper [15] the authors pay emphasis on need of trustworthiness of software to reduce the risks that the project faces. Although cost efficiency analysis and risk management are imperative to support the decision making process, it is less involved in the present software process model risk management. And there is a significant gap between the demand and the number of research on software trustworthiness measurement model. A software is termed trustworthy if it is able to satisfy the criteria of data security, data integrity, stability, and availability. It is tough to quantify trustworthiness as different users have different viewpoints and consider different aspects for terming a software as trustworthy. Software trustworthiness is related with risk control in the software process, as well as to the quality management of the software development process. Capability Maturity Model Integration (CMMI) level is shown to be have the maximum positive correlation with the risk identification and mitigation. The paper shows analysis done on various Chinese Enterprise in which 200 questioners were prepared for project manages. It was found that more finance and increased expenditure on human resources helps a software development company become more trustworthy. These factors also help in reducing the number of risks.

### III. CONCLUSION AND FUTURE WORK

The paper considers various software risk management techniques proposed in the latest years. As over the years the complexity off projects being made increases so does the risks involved in them. Without application of proper risk mitigation techniques, a project may fail. Over the years emphasis has been given to software design and development



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijircce.com](http://www.ijircce.com)

Vol. 5, Issue 4, April 2017

phase and not much on the risk analysis phase. As the results obtained in the above mentioned papers show that giving importance to risk analysis not only helps in decreasing the number of failures but also help in reducing the overall time taken to produce the software. Depending on the model of development of the software the risk analysis process is chosen. Evaluation of each component based on the risks it carries is very important. A particular functionality can be removed from the project if the risk involved it in outweighs the profit from including it in the software. Every risk is considered on the basis of probability of its occurrence and how much damage it will do if it happens. On the basis of this we decide which risks need to addressed first. Collaboration among employees is also listed as one of the aspects that may help in identification of risks. A freer environment allows everyone to present their viewpoint and helps figuring out risks. Another way is getting the feedback straight from user by doing a usability tests. It allows developers to know the importance of each and every component of their software and they can focus on the ones that are more important to the user. For projects which are being designed at various centres by different developers, effective communication plays a vital role in risk identification and management. A software that can be accessed by more number of people is more vulnerable to security attacks as happens in the case of websites. Such projects require extra emphasis on the security phase. Each project has different phases that require extra attention and with the use of correct risk management technique we can easily identify these phases. Various cultural and social issues also play important role in the life cycle of a project, it is important that our risk analysis technique does take them into account.

## REFERENCES

1. Chandan Kumar and Dilip Kumar Yadav, 'A Probabilistic Software Risk Assessment and Estimation Model for Software Projects', *Procedia Computer Science* pp. 353-361, 2015.
2. Robson Maranhao, Marcelo Marinho, and Hermano de Moura, 'Narrowing impact factors for Innovative Software Project Management', *Procedia Computer Science*, pp. 957-963, 2015.
3. Miguel Wanderley, Julio Menezes Jr., Cristine Gusmao, and Filipe Lima, 'Proposal of risk management metrics for multiple project software development', *Procedia Computer Science* 64, pp. 1001-1009, 2015.
4. Zvonko Kremljak and Ciril Kafol, 'Types of Risk in a System Engineering Environment and Software Tools for Risk Analysis', *Procedia Engineering* 69 pp. 177-183, 2014.
5. Cristina Lopez and Jose L. Salmeron, 'Monitoring software maintenance project risks', *Procedia Technology* 5 pp. 363-368, 2012.
6. Lazaros Saigiannidis and Prodromos D. Chatzoglou, 'Software Development Project Risk Management: A New Conceptual Framework', *Journal of Software Engineering and Applications* 4, pp. 293-305, 2011.
7. Paul L. Bannerman, 'Risk and risk management in software projects: A reassessment', *The Journal of Systems and Software* 81 pp. 2118-2133, 2008.
8. Haneen Hijazi, Thair Khmour, and Abdulsalam Alarabeyyat, 'A review of Risk Management in Different Software Development Methodologies', *International Journal of Computer Applications* 45, 2012.
9. The Westfall Team, *Software Risk Management*, Linda Westfall, 2010.
10. Nidhi Sehrawat, Neha Munsri, and Mahak Jain, *Risk Management in Software Projects*, *International Journal of Computer Science and Mobile Computing* 3, pp. 845-849, 2014.
11. Rajesh Garg, Nitika Bansal, and Pallavi Sharma, *Software Risk Management*, 2014.
12. Salma Firdose, *Investigational Analysis of Existing Risk Management Models and Prior Research Contribution: A Review*, December 2014.
13. Ayad Ali Keshlaf, *Risk Management for Web and Distributed Software Development Projects*, 2010.
14. Christin Lindholm, *Introducing Usability Testing in the Risk Management Process in Software Development*, 2011.
15. Minglu Li, Jianping Li, Hao Song, and Dengsheng Wu, *Risk Management in the Trustworthy Software Process: A Novel Risk and Trustworthiness Measurement Model Framework*, *Fifth International Joint Conference on INC, IMS and IDC*, 2009.