

# Risk Assessment Approach for Software Development using Cause and Effect Analysis

Abdur Rehman Riaz<sup>1</sup>

Syed Mushhad M. Gilani<sup>2</sup>

## Abstract

In the industry of software development, the risk is most effective competitor that tries to flop the project at any stage of development. Risk is a critical factor that is obscure and has the potential to wreak massive loss of the project in terms of money, time and resources. It is also harmful to the credibility of the organization. Most of the organizations don't focus on this factor, and as a result, might witness the project failing. Lack of risk assessment is very common in most organizations. This paper introduces risk assessment factors and analyzing the various situations to tackle these. We investigate the most effective key risk variables cost, strategy, technique, operation, and some unknown/unpredictable factors. Based on these variables, survey and interviews are conducted and examined. We applied empirical studies on these variables and map them on the cause-and-effect analysis technique. The proposed technique elaborates the factors behind these risk variables. After that, results and analysis of these variables have been incorporated to scale down the impact of risk.

**Keyword:** Software Project Risks, Risk Management, Risk Assessment, Risk Analysis, Cause and effect analysis

## 1. Introduction

Threats are identified during the risk analysis phase that may harm project development. Risk ensues from two factors: prediction of project failure, and its impact on the project [1]. Risk can be of any type like cost, time and scope. Risk analysis is a critical task that needs detailed information about the project so that these issues can be resolved and save assets of the project. During the planning of software project development, some risk analysis methods are used; such as qualitative risk analysis which is applied to the project for tracking down the issues that affect the quality of the project. Software project management is a tricky task to handle project [2] because the whole project depends on software project management. It helps to get profit or loss and demands a lower cost, minimum time, higher productivity, good quality and fine customer satisfaction to deliver the product right time in the market. Risk management is one of the vital part of software project management due to that risk is involved in each software engineering process [3].

---

<sup>1</sup> PMAS Arid Agriculture University, Rawalpindi | [abdurrehman.ar475@gmail.com](mailto:abdurrehman.ar475@gmail.com)

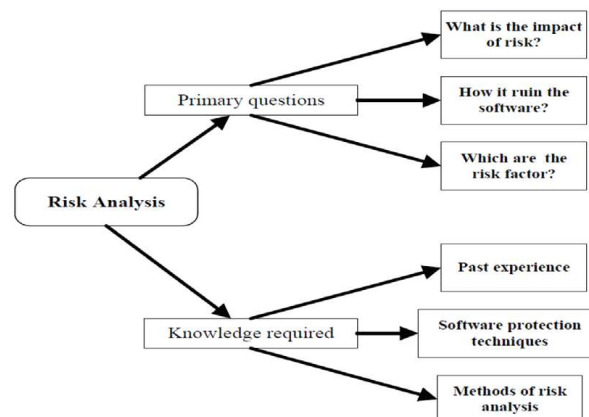
<sup>2</sup> PMAS Arid Agriculture University, Rawalpindi | [mushhad@uaar.edu.pk](mailto:mushhad@uaar.edu.pk)

The basic rule of risk management is defined in ISO 31000 as a state that risk management can create and protect value [2].

In every phase of the project, there are chances of risk, and it is the biggest challenge in software development to remove the risk from these phases. Ignoring the risk factors in the project can lead the project towards failure. Project failure is a loss of time and cost that spoils the image of the developing organization. In the past, many risk mitigation factors identified, but with time, the way technology increases, threats of risk are also increases. So there is always scope for identifying risk in software development. Risk is inherited in every project at every stage of the project. A good analysis of risk plan makes a software project effective and efficient which can fulfill all requirements of the project.

A basic taxonomy of risk analysis is shown in Figure 1. Before doing a risk analysis of a project, some primary questions arised. The risk analysis approach can answer these questions. Past experiences are a proactive approach for tackling risks and you get insight from it. Risk analyst has a knowledge base about the techniques to protect software like hardware base protection, adding watermark, checksum, cryptography for protecting data and guards [4]. Knowledge is needed about the method when risk analysis methods are applied to some projects. The method will be qualitative or quantitative, depends on the type of project [5]. Keeping in mind all the taxonomy, this research provides an approach that made the risk analysis more reliable.

This research paper puts attention on software risk factors. Widely affecting risk features are studied and perceive the four most harmful factors from them. These factors are useful insights for project managers while developing the software. A detail debate on practical problem and practical solution of an application were discussed by using these factors. Empirical studies are done by surveying, and results are extracted. The final results show that what's the main source of these four key risk factors and how to tackle these types of risk. The impact of each one risk factor is also discussed so that the project manager handles these risks based on their consequences respectively.



**Figure 1: Basic taxonomy of risk analysis.**

This paper has further divided into five sections. Section II describes the literature review and what studies are discussed in the past. Section III describes the methodology that how the collection of data can be used qualitatively so that risk analysis has been proved mathematically. In Section IV, results have been discussed and analyzed how these improve the risk assessment. In the final Section V, the paper sums up the whole study and discusses the conclusion and future work.

## 2. Review of Literature

The reason behind the delay in the completion of the project is mostly the lack of risk management techniques [1]. There is a need to minimize risk from every aspect of the software. Every software has some outcomes which are based on the customer requirements and market needs; which are difficult to handle both at a time. There is a need for empirical study time focus and managing strategy in every process of software development [3]. The studies at [2] have a systematic review type paper and have some limitations that the biases of interviewers and get the bad result from them. There is also a limitation of the area. This survey is done in a city of Denmark so every area has its specific pros and cons. Previously, many research papers published in the domain of software risk analysis. These papers also made a comparison between the different approaches used in the older papers. There is the use of fuzzy logic base modal for the aggregative risk management in the filed software development. It is an easy approach, but it has missing with the use of earlier models [6].

The software industry is based on the management of development. Risk management is one of the challenges of development. In [7] given the way to lessen the risk by using success parameters cost, time, people and process. According to the authors of [8], an architectural approach is used for risk analysis and alignment with agile development; and very few works implemented on it. Traditional risk assessment techniques like qualitative risk modal is bulky and inflexible. From the author of [9], who was inline qualitative risk tool with the phases of the software project. There is always a need for a systematic risk approach in project development. In the paper [10] researchers focus on the root cause of the risk factors which show a negative impact on the project. A fishbone technique is used in it and also conducted a survey to get the results both qualitatively and quantitatively. There is a need for the countermeasure to evaluate the risk control process.

Now a days Agile and its methods are mostly used for software development. As Agile support changes in any stage of the development so chances of risks also increase. Tools were proposed for minimizing the risks in agile project [11]. A framework is presented in [12] named RIMPRO that focuses to connect the product owner and manage its activities to reduce risks. A mathematical model is used for the prediction of risks and this model

is implemented in some real-world scenarios [13]. The prediction is based on similarity analysis of the projects [14].

A model was proposed for the assessment of risks in the phases of software development life cycle and proposed model was also analyzed on SPSS software [15]. A linear programming approach was used for the risk analysis of the project. Risk is a vital part of the project. The team experience is the most effective part of risk management. As risk management is vital, it completely depends on the maturity of the team [16]. Eleven Risk factors are most important and they are the root cause of all the factors. New technology complexity changes incompleteness and ambiguity are the most common. These all factors are an obstacle to the success of the project. Poor planning and bad commitment from the customer are external factors that increase the risk of the project [17]. For increasing the development procedure expelling risk factors in the early phases of development. Risk management was done at the stage of the developed premise of risk possible to happen. The expectation of risk is from the programming designer [18].

**TABLE I. Overview of Literature Review**

Paper	Contribution	Limitation
[1]	Empirical analysis on selected process modal	Have not a focus on specific risk factor
[2]	How stakeholders perceived important the value of PRM	PRM is a broader term that is not completely covered
[6]	Use Fuzzy logic to evaluate and minimize risk in software	This logic is not used with earlier approaches
[7]	Risk propagates with ripple effects is not identified and eliminated.	The impact of risk on major project success parameters such as Cost, Time, People and Process.
[8]	Risk assessment and management of Agile project by using qualitative tools	Lack of appliance in industry and real projects.
[12]	RIMPRO framework was presented for managing the product owner role	The presented model was not applied to some real project
[16]	Most and less common risk management practices in Scrum projects are identifies	Need to adapt classic risk Management and level of risk aversion, and integrate it with the projects
[17]	Analyzing the 11 most mentioned factors. Relevant most is requirement risks	No assessment of these factors, seeking to reinforce the indicated results
[18]	Introducing a risk assessment step that can be automated	Automated modal cannot link with some tool.

### 3. Methodology

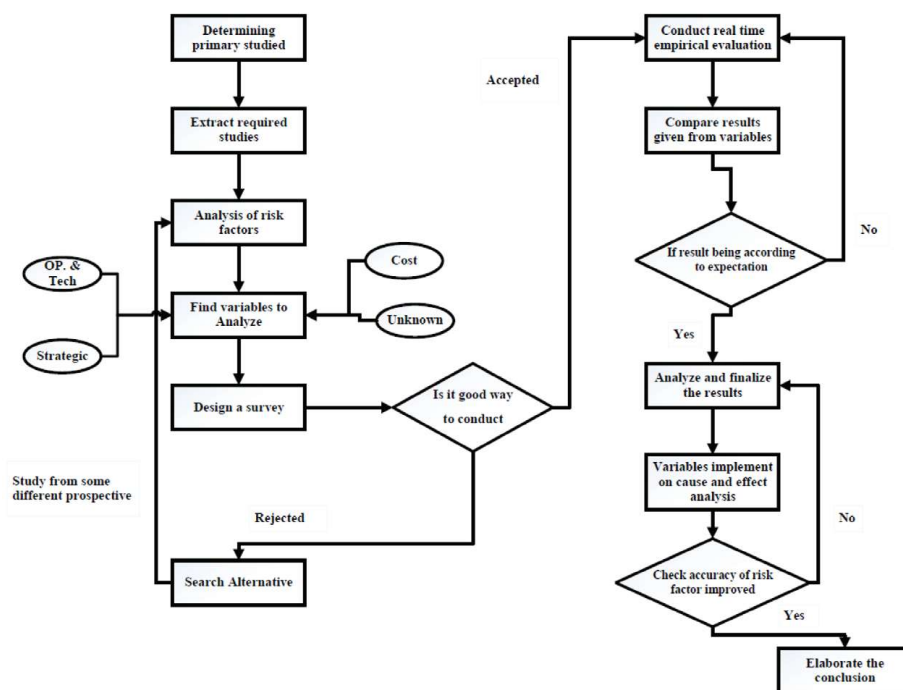
In our study, we have to find risk assessment factors so that it's made better understanding to risk. The basic hypothesis of this research is finding the dependent factors which are



the cause of independent risks. Several research papers on the topic of risk analysis, and risk management are studied. After analysis, we observe that risks are coming from starting level. For solving this problem, cause and effect analysis is used which helps us to understand the root causes of risk. A survey is also conducted to get the data from real-world and developers who practically work on projects. The proposed work diagram is shown in Figure 2.

For analyzing risk factors, a survey is conducted via questionnaire 1. The questionnaire is shared in different software houses in Pakistan and also send to some other countries where we have our references. The questionnaire was solved by senior developers who had a minimum of 3 years of experience in the development and management of the project. The questionnaire was filled by all types of developers like mobile applications, desktop applications, hybrid applications and web applications. The questionnaire is designed online on Google form so that it can be shared easily in the situation of the pandemic of Covid-19. This questionnaire was shared with almost 18 companies, and 32 developers from 14 companies responded.

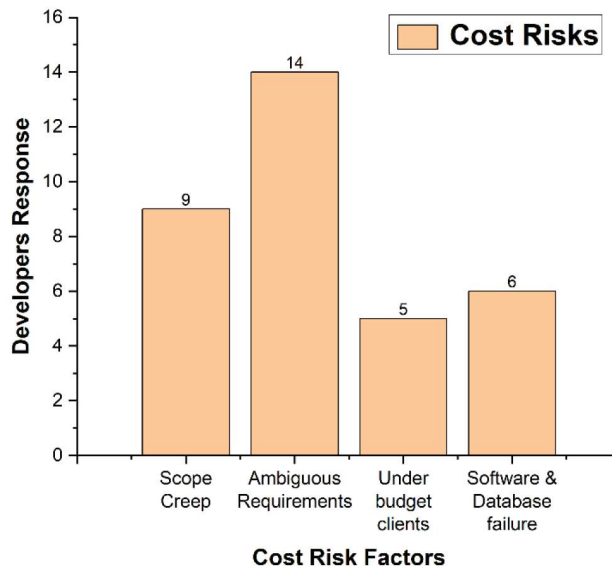
The focus of our research is based on some variables that are the key cause for the risks in the project. From studying the literature, it was found that those variables are risk-related, cost, strategy, operations & technique and some unknown risks. The survey is based on these variables. The meaning of the term “unknown risks” is varying from one developer to another. Here is the discussion of responses from the survey.



**Figure 2: Work flow diagram for the proposed risk analysis approach**

### A. *Cost Risks*

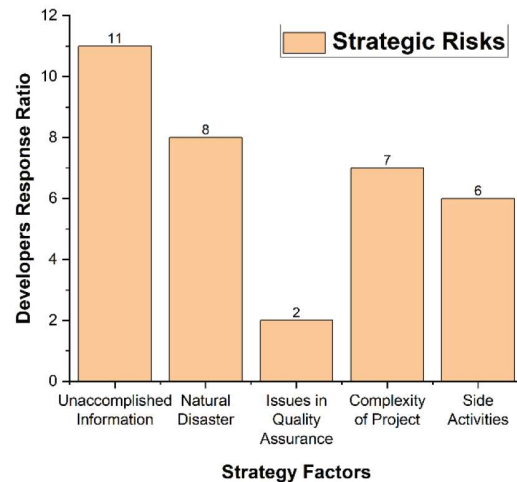
As Figure 3 represents that in the response to cost risks questions, most developers think that requirements are not clear and sometimes there is a fault in the machine which can increase our expense. Scope creep is another big reason because changing requirements is a cause of the effort-loss of the developer. Sometimes the client has a low budget and it is difficult to manage the project at a very tight cost.



**Figure 3: Cost Risks Empirical studies response chart**

### B. *Strategic Risks*

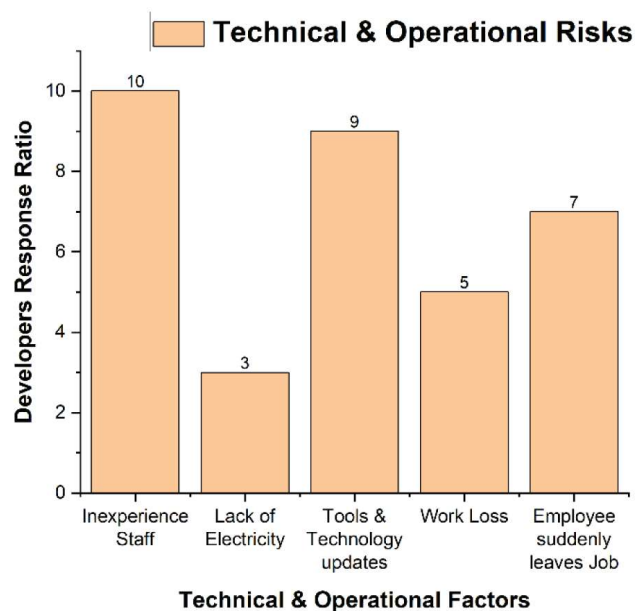
The questions linked to strategic risks tell us that information given to the developer was not complete so that could disturb the schedule. There are recurring issues in quality assurance. Sometimes the project looks simple at initial level, but in reality, it is complex due to which entire strategy becomes ruined. Some unexpected circumstances like coding errors could not be solved in the expected time which increases the duration of the project as illustrated in Figure 4.



**Figure 4: Strategic Risks Empirical studies response chart**

### C. *Technical & Operational Risks*

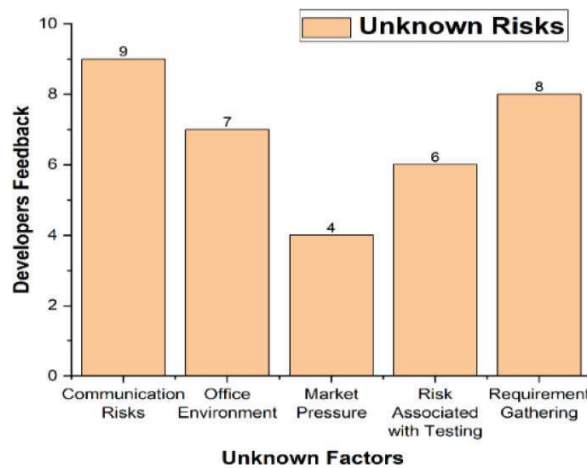
In this type of risk some technical type of risks occurs, like loss of all the program due to crash of hard disk or any drive. There are also network problems: the server become down or any cable is damaged and the project is disabled to save to the cloud, code is not saved due to the shutdown of electricity, and entire work lost. In some cases, an employee leaves the job in the middle of the project and all schedule of the company comes into the trouble. Lack of group work and unprofessional techniques also disturb the schedule of the organization. User interference issues also create risks for the project as shown in Figure 5.



**Figure 5: Technical and Operational Risks response chart**

### D. *Unknown Risks*

These are the risks which developers identify from their experiences. These risks do not have specific name or definition, and these change from company to company and developer to developer. From our responses, we get the names of risks like market risks, communication risks, risks related to office or organization, risks from the inexperienced staff and how requirement gathering and architecture are designed that increase the risk exponentially as shown in Figure 6.



**Figure 6: Unknown Risks Empirical studies response chart**

## 4. Results and Discussions

Here are the results of the responses, we got and the analysis we did of those results with cause-and-effect diagram, also known as fishbone diagram. Our research gathers some statistics from the real-time environment of software development. We implement our responses on graphs for more understandable for software engineers. Finally, we analyzed the results of all variables as discussed above. These results are shown by cause-and-effect diagram and make it easy for the software industry to implement them on their projects and mitigate the risks. Overview of all risks is drawn here in a graphical form that shows which risk is more impactful. Figure 7 shows the graphical representation of all the risks.



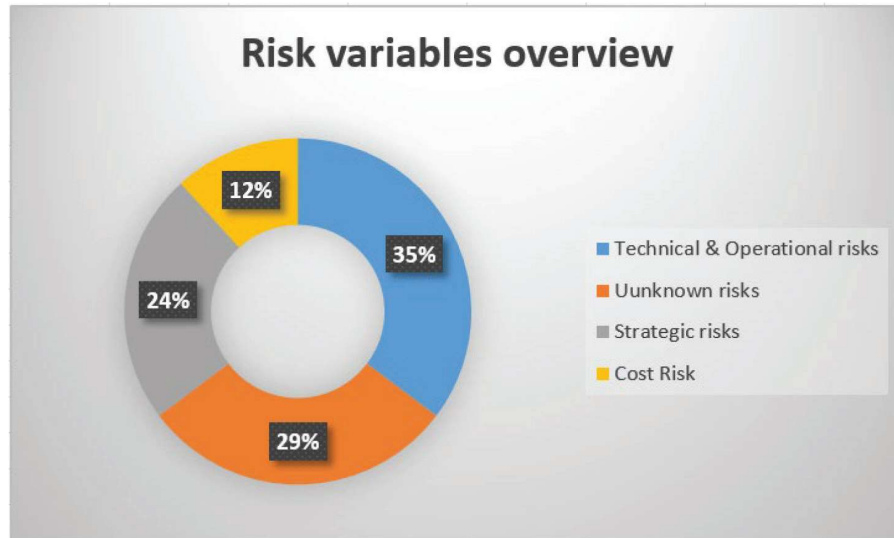


Figure 7: Risk variable overview in regarding their impact

#### A. Cost Risks

It is one of the basic risks of the project after doing a survey on it and analyzing its root causes which can enhance it. First of all, scope creep is the main cause of client changes requirements time by time. These changing requirements waste a lot of time and money. Time and money are directly proportional. These are waste in parallel. In the same way, ambiguous requirements are also a cause of this. Machine and software crash is also a loss of money. Figure 8 shows the cause and effect diagram. The solution is to always make a copy of the project so that in case of a crash we can save from big risks. In some cases, the client has a very low budget but he/she wants software with all functionalities. To avoid this type of fatigue we have to avoid those clients. Because working on a low budget can create tension in the mind of developers.

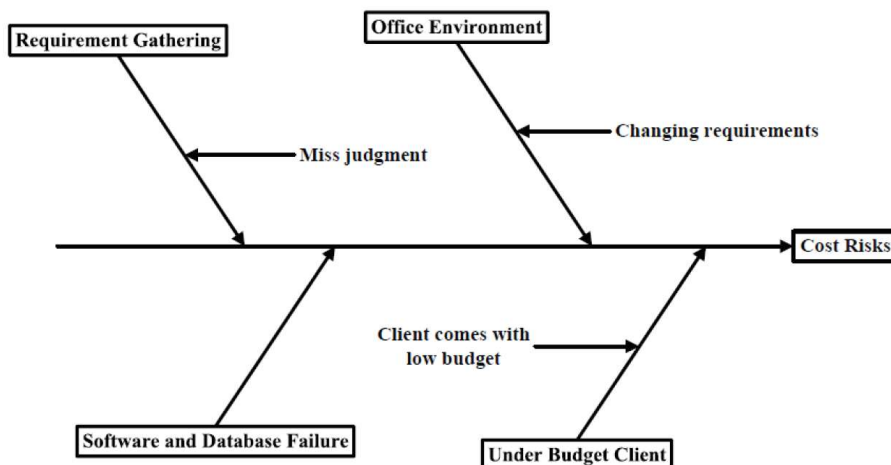


Figure 8: Cause and Effect diagram for Cost Risks

### B. Strategic Risks

There are always chances of changing plans. That’s the way every reputed organization has more than one plan. Incomplete information was given by the client; stakeholders or developers can create schedule changes. Before starting a project, a clear concept of the software is needed. Those companies who cannot focus on the main project and give time to other projects whose deadline is very far, also have a problem with the schedule. The company needs to have an eye on the project deadline. A very main cause we found on analyzing the cause and effect diagram is natural causes. Sometimes, the developer sees the sudden death of his relative or experiences some illness due to seasonal changes. The organization has no control over this type of cause which is why space for natural causes is always needed. One more cause from the developer’s view is that some projects look simple, but they are very complex and disturb all the strategies of the company as represented in Figure 9.

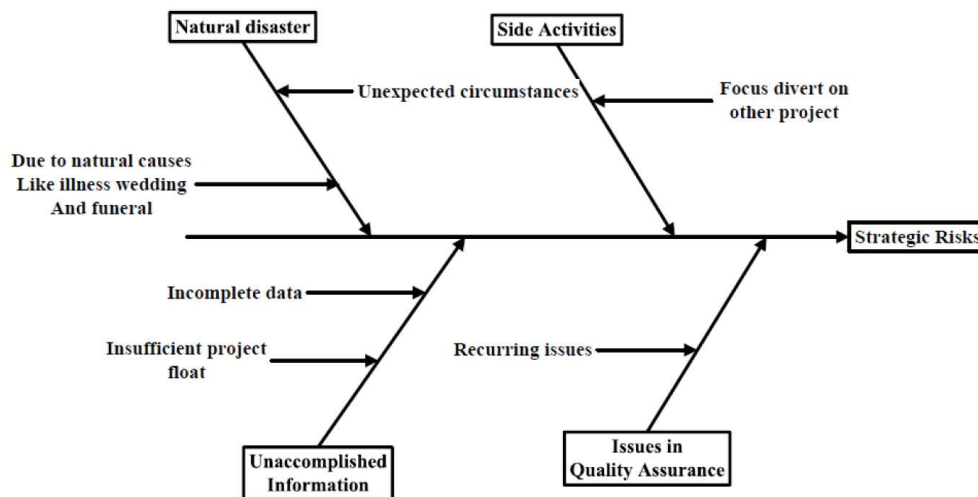
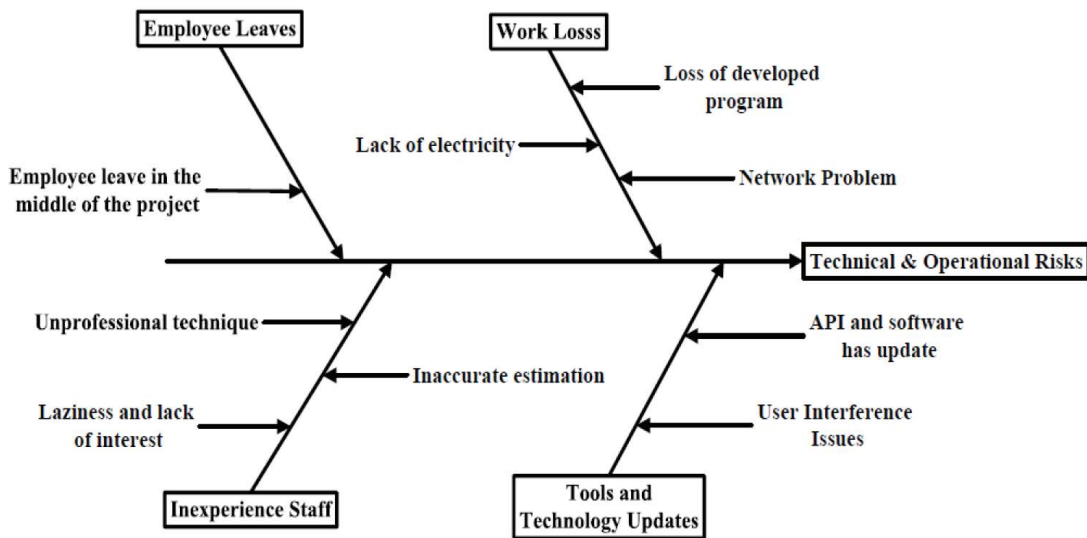


Figure 9: Cause and Effect diagram for Strategic Risks

### C. Technical & Operational Risks

The cause and effect diagram shows that technical and operational risks are very impactful for the software. As shown in fig. 6 it has a 35% impact on overall risk factors. In an organization, every employee is a pillar of the company. If an employee leaves the job all operation of the company is out of order. There is a need for an agreement with an employee so that he cannot be in the middle of the project. Due to network problems, lack of electricity and burning of the machine can cause a technical problem for the software. The project always needs a backup to overcome these risks. For working on the main modules of the project, placing experienced staff are optimal solutions. The inexperienced employee can leave some bugs in the project. These are not good for the reputation of the company as shown in Figure 10. Tool and technology usage in software is upgraded

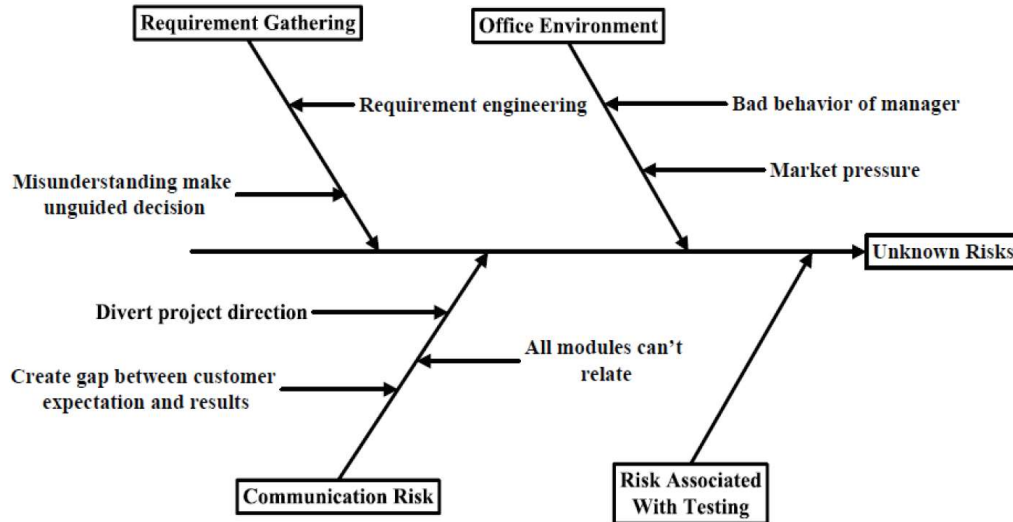
day by day. Technology has innovation every day. Devices and software are a big issue of compatibility. APIs, IDEs, frameworks and languages have a new update after a year. So it needs to use applications and tools which support every version or has some option of update. Because it's a very alarming cause of the risks of the project.



**Figure 10: Cause and Effect diagram for Technical & Operational Risks**

#### **D. Unknown Risks**

Different developers working on different languages have their own experience. Companies have their risks on behalf of their experience. After analyzing this and making a cause-and-effect analysis, most companies focus on the requirement engineering process. Mistakes in the requirement gathering process can cause big risks for the project. Some developers say that the office environment has caused some risks like bad behavior of manager, bad power system of the office, and there is no concept of group work. The manager needs to measure these things and improve on them by taking suggestions from employees and experts as depicted in Figure 11. Lack of communication is a fluently discussed topic in organizations. Developers have an opinion about that, it can divert project direction. Requirements are not clear if there is a lack of communication and ambiguous modules are made. It is very difficult to join these modules. It creates a gap between the stakeholder and developer, and the company cannot get expected outcomes. While testing the software, an experienced tester is needed. Who can test the software and tell the bugs to the developers in a clear way.



**Figure 11: Cause and Effect diagram for Unknown Risks**

## 5. Conclusion

The quality and success of the project are based on the risk assessment of the project. The focus of this research is on risk assessment, finding its root causes and analyze of root risk factors. The paper also describes the consequences of that risk on the project. Cause and effect diagram analysis is used for finding the causes and their effects of different types of risk. Our studies describe four risk factors that are faced by every software project. These four variables help software organizations how they can reduce risk factors, and they can also find causes that are based on this type of risk.

For future work, methodology can be extended and integrated with real-time applications to evaluate risk using these factors. Research presented here find causes and effects of risk factors in different software houses of Pakistan by changing the location results, these might be changed.

## Acknowledgment

The Authors elegantly acknowledge, "UIIT, PMAS-Arid Agriculture University" for their help and support.

## References

- [1] R. Bista and S. Karki, "A New Approach for Software Risk Estimation," 2017.
- [2] P. Willumsen, J. Oehmen, V. Stingl, and J. Geraldi, "ScienceDirect Value creation through project risk management," *Int. J. Proj. Manag.*, vol. 37, no. 5, pp. 731–749, 2019, doi: 10.1016/j.ijproman.2019.01.007.
- [3] R. K. Bhujang, "Analysis of risk in software process models," no. *Iciss*, pp. 199–204, 2017.
- [4] Memon, Jan M., et al. "A study of software protection techniques." *Innovations and Advanced Techniques in Computer and Information Sciences and Engineering*. Springer, Dordrecht, 2007. 249-253.
- [5] S. S. Muhammad, K. Weyns, and M. Höst. "A review of research on risk analysis methods for IT systems," *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, 2013.
- [6] K. Bansal and H. Mittal, "ANALYSIS AND EVALUATION OF SOFTWARE AGGREGATIVE RISK USING," pp. 172–176, 2014, doi: 10.1109/ACCT.2014.99.
- [7] R. K. Bhujang, "Propagation of Risk across the Phases of Software Development," pp. 508–512, 2018.
- [8] M. G. Jaatun, "Architectural Risk Analysis in Agile Development of Cloud Software," pp. 295–300, 2019, doi: 10.1109/CloudCom.2019.00050.
- [9] V. Anes and R. Santos, "A New Risk Assessment Approach for Agile Projects," pp. 67–72, 2020.
- [10] M. T. Riaz, "Risk Assessment on Software Development using Fishbone Analysis," 2019.
- [11] Tavares, Breno Gontijo, Mark Keil, Carlos Eduardo Sanches da Silva, and Adler Diniz de Souza. "A risk management tool for agile software development." *Journal of Computer Information Systems* 61, no. 6 (2021): 561-570.
- [12] S. Lopes, R. Gratão de Souza, A. Contessoto, A. Luiz de Oliveira, and R. Braga, "A Risk Management Framework for Scrum Projects," vol. 2, no. *Iceis*, pp. 30–40, 2021, doi: 10.5220/0010448300300040.



- [13] B. Gontijo, C. Eduardo, A. Diniz, and D. Souza, "Risk management analysis in Scrum software projects," vol. 26, pp. 1884–1905, 2019, doi: 10.1111/itor.12401.
- [14] A. S. Filippetto, R. Lima, and J. L. V. Barbosa, "A risk prediction model for software project management based on similarity analysis of context histories," *Inf. Softw. Technol.*, vol. 131, no. December 2020, 2021, doi: 10.1016/j.infsof.2020.106497.
- [15] Nayak, Malaya K., and Arka K. DasMohapatra. "MANAGEMENT OF QUALITY IN IT PROJECTS: A RISK ASSESSMENT MODEL."
- [16] K. Adimora, "Application of Linear Programming for the Optimization of Software Risk Assessment Model ( Osram )," vol. 5, no. January, pp. 118–126, 2021.
- [17] J. Menezes, C. Gusmão, and H. Moura, "Risk factors in software development projects : a systematic literature review," pp. 1149–1174, 2019.
- [18] G. Nasreen, M. Asam, and S. Afsar, "Automated Risk Analysis Software Model for Enhanced Software Development," vol. 28, no. 6, pp. 5197–5200, 2015.