Review Study on software risk factors

Ahmed Saleem Abbas 1, Ashwak Alabaichi 2, Maisa'a Abid Ali Khodher 3
1 Software Department, College of Information technology, University of Babylon, Iraq
2 Biomedical Engineering, College of Engineering, University of Kerbala, Iraq
3 Computer Sciences Department, University of Technology, Iraq

Ahmed_saleam@yahoo.com

Abstract. Risk management is an important aspect of any software project management. Risk management conditions are a position that prevents the software project from be a successful and within time or intervenes to make a project late. This is a critical aspect of software product development, where it is often ignored. From that, software project development schedules are negative influenced by managers themselves pressing their employees to overcome time losing problem when it was planned to be properly planned. Each software project faces risks of all types and seriousness. Design flaws nearly cause 50 percent of security problems, so the analysis of risk at the design stage is very important issue of the software security. Risk analysis is also a means of elicitation the required data to make a good decision depend on knowledge about sensitivity parts, threats and impacts. Most of the risk analysis methodologies share some common principles and limitations but also have advantages and disadvantages when implemented to recent software design.

1. Related Works
In 2007, Noor Habibah Arshad and et. al. These researchers said in their published paper that the increased complexity of software development management has a significant impact on risk factors within the software development area. When working on a project development, project leaders must have knowledge of all the apparent and anticipated risks that may affect the completion of the software project. Through this research a group of risk factors have been reviewed that may lead to the failure of the project in whole or in part. The results showed that communication in the development environment of the software project among its employees was bad communication and that this poor communication is often due to being poor listeners. In order to minimize failures, there must be an improvement in the communication between the team working on a joint project. Thus, the results of this research can play a significant role in helping project managers to integrate risk factors into the software development methodology. The main objective of this research is to identify important risks that have an impact on software development in the public sector. From this research, it is clear that poor communication between the department members and the team members is one of the most important factors that led to the failure of the software project. Poor communication is one of the most important problems facing the project. The researchers recommend working to enable bilateral communication between team members in order to praise this problem in the future. [1]
In 2008, Paul L. Bannerman published articles and said that the success of any project, especially software projects, requires control of the risks that may face the stages of implementation of the project. Through this research, the researchers reconsider the risks, especially through practical application and published theories. This study is based on the practical application and documented practices of Australian government agencies, where they have worked to find problems and gaps within the public sector. They found that the risk management process was low in performance compared to the expected losses. There is a challenge in the research results of some traditional concepts of risk management and project management. There are many examples to mention, in software projects there is no uniform structure and as in many other projects. This leads to significant differences in the nature of project management and the nature of the risks facing their projects. The researchers pointed out that the main management of the project is not enough to ensure the success of the project. Researchers concluded that risk-based research requires a lot of experience and practice, and that the risk management can sometimes slow down projects progress. The researchers found that the concept of risk as a threat of negative impact is highly relevant to software projects, and that there is an urgent need to manage such threats in order to achieve the desired results of the project. They found that risk management varied considerably between theory and practice. Failed software projects cause a lot of cost losses, so researchers and experts must continue to learn from each other to reduce failures and develop new ways to lead to successful projects. [2]

In 2009, Robbie T. Nakatsu, Charalambos L. Iacovou Researchers worked on risk factors related to external software development. They had several objectives, including the establishment of guide lists of risk factors for both external and local projects. Comparisons between these two types of projects and identifying the most important and dangerous risk factors for each of the two types, in order to identify the important risk factors from the client's perspective, two Delphi questionnaires were applied, both in the exterior and interior areas. The researchers made a qualitative comparison to identify differences and similarities in the results of surveys across their risk profiles. They found three types of risk: those that appeared in both types, but increased in the outer type; those that appeared in both types; and those that appeared in the outer type only. Researchers also pointed out that traditional project risk management is the most important in both types; however, the external type may be more susceptible to some traditional risks as well as rare factors of its kind. They identified the most important risk factors but did not discover appropriate ways to manage and overcome them. Another limitation is that guided profiles are project management. This result happened for two reasons. First, members of the IT management team, selected based on their project management experience, may focus on these concerns. Second, this research study relied on published project management research, which was used as a starting point in generating risk data. As a result, outsourcing, particularly related to long-term effects, has not been discussed. The third limitation is that this research needs more theoretical side. This study has an exploratory nature and is an attempt to identify risk factors. The fourth and final determinant is that the comparisons were of a qualitative nature between the two policies. [3]

In 2009, Ozren Đurković, and Lazar Raković The risk is an integral part of everyday life, the researchers said. The process of developing information systems is a complex process, which leads to exposure to a large number of risks. Many software projects are not able to achieve the predetermined goals, so the development of information systems risk management is indispensable. In this work, the effectiveness of the developed information systems is checked, and a number of risks that may affect the development of the software project are also investigated. Also, the researchers directed their attention towards the methodology for risk management, and a risk management framework was developed simultaneously. They concluded that most information systems development projects are at risk and that this factor is a must, so it is very important when developing projects to take these risks into account. Many projects are either incomplete in terms of objectives or have not been completed in aggregate. It is necessary to define a risk management methodology to identify all project risks. It is important to identify management risks using certain methods that depend on the likelihood of impacts, their perceptions, and the costs of risk. Therefore, it is necessary to develop a risk management plan and it is important to supervise the implementation of this plan, as well as to identify and conduct tests
periodically to current and new risks that could adversely affect the development of information systems. [4]

In 2013, Srikrishnan Sundararajan and et. al. They state that: This is an empirical study of the practice of risk management in the software development industry. A revolution in information technology (IT) has been announced which is witnessing clear changes in the practical methods and methods of software engineering and IT business models. These events led to changes in risk management and to the perception of the nature of these risks. Although there is a lot of research on the risks associated with software development, there is still a real need to create a useful and practical software risk management model. This model was proposed by researching contemporary industry practices. Based on the analysis and study of 145 software projects, a risk management model has been proposed and the specific features of this proposed model compared to the current work related to the program risk have been clarified. This model identifies the main areas of focus for risk management in offshore projects in order to provide the best results. They concluded that: There is little published research on software risk management and associated industry practices. Over time, software risks and ratings have changed with (1) an IT revolution in terms of hardware and software components, (2) significant development in software engineering and (3) deterioration of ownership of software project management. There is a significant difference between software risk, risk classifications and risk management, in the context of software outsourcing or offshoring or company-owned software development. Although mainstreaming software risk management is a very important area of research, the search for existing IT models, such as outsourcing, offshoring, and distributed project management would add a lot to IT and significantly in practice. There is a need to explore differences in practice across many project categories, for example, software migration, maintenance, update, COTS. [5].

In 2014, Haneen Hijazi, and et. al. They said that: Each stage of the software development life cycle (SDLC) can be exposed to different types of risk factors. Identifying and understanding these risks well is an essential stage for successful risk management. Through this research, a complete theoretical study of the main risk factors which is considered a threat to each stage of SDLC. A comprehensive list of 100 risk factors was announced. This list contains the most frequent risk factors that threaten most software development projects. In this paper, this checklist has been developed, which is a checklist that can guide the project team in identifying potential risk factors and assisting them in designing strategies to mitigate or avoid these factors. [6]

In 2015, Ewa Ziemba and Iwona Kolasa An article aimed at answering a question about risk factors for information system (IS) projects in public institutions in Poland was published. These factors were known to support practical collaboration, case study, logical conclusion, and literature review. The procedures were as follows. First, explain the relationship between project success and risk factors and highlight the risk factors in the literature. Second, the risk factor screening technique is awarded to public and private organizations of the Information Systems Project. Third, the risk factors for information systems projects in public organizations in Poland are identified, and the risk factors presented in the literature have been improved. Within this framework, the Square Factor is measured in eleven group, for example: support for senior management, management of operations in organizations, involvement of end-users and so on from other teams. We conclude from this research that identifying and understanding risk factors is critical to the success of information systems projects in public organizations. It has also enhanced the risk factors identified in the literature and has proposed a comprehensive risk factors framework for information systems projects in public institutions. This study employs two methods to identify risk factors. First, risk factors for information systems projects are analyzed and presented in the business organization. Second, the unique risk factors for information systems projects in public organizations are identified on the basis of a case-study approach. This research can be useful for Central and Eastern European countries. This is because these countries are similar in terms of geopolitical situation, history, traditions and common culture. In addition, the similarity is reflected in the construction of a free market economy and democratic state structures, participation in the process of European integration, and the level of progress in the implementation of information systems in public organizations. Moreover, they must solve the same problems and
overcome the same economic, political, technological and social obstacles in their transition from traditional public organizations to information-based organizations. This study has a number of limitations. This research is based solely on two studies in Poland, so caution should be exercised in generalizing the results and the risk factors for information systems projects in public institutions should be further explored. Further studies are needed to verify and strengthen the risk factor framework. [7]

In 2015, Maruf Pasha and et. al. For a long time, researchers have focused on risk management systems in software development. One of the most important tasks within software engineering is software risk management, which in turn contains identification, assessment, mitigation and control of these risks. It will produce a disciplined environment that contributes to efficient decision-making when assessing potential problems during software development. In large-scale and complex systems, there is a difficulty in measuring risk, as there are many risks that can arise in such systems, so it is a very important challenge. There are relative differences in risk factors between complex and small systems. Through this research these differences are described and a comprehensive list of risk factors is announced. The researchers conducted a detailed comparative analysis of the various risk management models associated with the software with some of the common features identified and then classified into several categories based on their risk severity. They found that the most important thing in software development is to carefully assess and control risk factors. In recent decades, many risk management techniques and support tools have been proposed. Intense competition between organizations forced them to use these techniques to improve product quality and risk assessment. Thus, this study aims to provide a critical analysis of these various risk management techniques. This analysis helps other developers choose the best technology, model, or framework suitable for their software project. A comparative analysis table that helps analyze risk management models in comparison with previous studies has been illustrated and some common features and factors have been recognized and have concluded that each model has different specifications that may be applicable in different environments. [8]

In 2015, Ungureanu Anca, and et. al. the researchers said that the risks faced by the project is an uncertain event, when it occurs may have a positive or negative impact on one or more of the objectives of the project, as an example of these objectives: delivering the project within a specified schedule and within a certain level of quality at a specific cost agreed in advance. There may be aspects of the organization or project under risk conditions, such as immature project management practice, reliance on external participants who directly control the project, lack of integrated management systems, and competing projects. There are a variety of definitions of project risk management arising from the fact that each project may have specific types of risks that may affect project outcomes and in varying proportions such as construction projects, IT projects. They concluded that project risk analysis is an effective way to ensure that the strategies used to control the potential risks of the project are profitable. To develop a risk management plan, the starting point is to analyze the risks in the preparation phase of the project. By following the steps and activities to be applied in the risk analysis model, it is possible for the project manager to lay the foundations for implementing the risk management strategy implemented during the project life cycle. Once the analysis is completed and the potential risks identified and assessed, the project manager should establish a risk management team with responsibility for addressing risks and implementing mitigation strategies. [9]

From these related work we found that there is no reliable published dataset contain practical features of software projects development, so it is necessary to create like this dataset to support software developers in risk analysis and risk prediction by using Data mining technique.

2. Introduction

The program's major risk assessment and simple assessment can be isolated by its capability to implement classic risk identification to program design and thus produce correct simple requirements. The approach to analyzing repeat risk must be integrated deeply through the SDLC (software development lifecycle) [10].
3. Traditional Risk Terminology

Many methodologies in risk analysis that detect risks from different aspects. Some of basic approaches are [10]:

3.1 Financial loss method:
Which look to produce a loss figure to make balancing in the cost of implementing different types of controls.

3.2 Risk ratings:
Conducted mathematically for risk measurements with arbitrary rates of threat, impact and probability.

3.3 Qualitative assessment methods:
Where the assessment of risk depend on story telling or factors of knowledge. Each method has very different parts but there are common concepts that must be available in most of risk analysis.

3.4 Threat:
It is an aggressive threat to the security of the system because of a financial objective.

3.5 Vulnerability:
It is a weakness or defect in the any step of SDLC of system security that can be compromised by any attacker.
Figure 3: The type of Weakness points

So there is a need to predict the number of weakness points, (Jinyoo Kim; et. al.) modeling the Weakness points detection process, depend on the measurements of source code sharing among software systems that had multi-version. Like this modeling, it can be useful in evaluating security risk of new version before and after its release. Countermeasures: are the management, operational, and control to prevent the risk. Alternatively, minimize the expected risk [11]. Example: (the risk determination)

1- A function to compute the probability of event and loss.
2- Checklists contain weakness pint, vulnerabilities and threats that used to define a basic measurement of risk.

4. Estimating the exposed to risk

There are two estimations method in risk exposure analysis:
   a. The size of potential time loss from a predefined risk factor.
   b. The probability of actually loss occurrence.

The function (1) is a basic formula to determine the calculation of exposing to risk:-

\[
\text{Risk exposure factor} = (\text{Percentage of loss probability}) \times (\text{loss Size in weeks}). \quad \ldots \ldots \ (1)
\]

For example, at one major financial organization, there were a development project that the author worked as a main worker from the start time of project development; probability have to be assumed as a 60% of some impact on a project by delays from technical support. If a conservative estimate of a loss in time assumed to be approximately 3 weeks than the risk occurrence would be calculated according to equation (1): 60% \times 3 = 1.8 (weeks).

The result is 1.8 weeks, where it is represent a risk exposure factor of the possible loss. Since the manager are estimating a 60% probability of this occurring so they are not expecting to lose the complete three weeks in time. The Estimate process of the loss probability is much hard than doing the same for the loss size. There are several activities can be done to enhance the accuracy of this part of the equation [12, 13].

- The person have very good experience in (the development environment of the system and it’s political infrastructure).
- For each risk estimates the probability and then make a risk-estimate review.
- Apply the “Delphi” method; in this method, for each risk factor all the project team-members make their own estimation individually. Then discuss the integrated risk-estimate reviews to define the most near probability of each risk factor until reach to the satisfaction of the all team member with a final analysis.
- Apply “easy betting analogies” with significant amounts of personal money. For example, if the required means, contrivance, and instrument were available and preparation on time you would earn $150.00, if they were not, you would lost $100.00. The probability of risk is the dollar amount on the downside divided by the total amount of dollar,

\[
\frac{100.00}{100.00 +$150.00} = 0.40.
\]
- Apply “adjective calibration”; each team member should select a risk level from “highly likely” through “highly unlikely” in terms of a verbal scale of phrases. Then make conversion from the verbal assessments to quantitative assessments [14].
5. Common Themes
The risk analysis process is an ongoing process that cannot be represented as a simple step that is completed at one stage of the SDLC of the software product; also, this process focuses on identification, classification and reduction of risk. The produced results of risk categories and risk analysis are very connected to both requirements and test procedure, where developer can use these results to identify and make a plan of specific tests. Good risk analysis is highly dependent on understanding the work impact so it is necessary to understand laws, regulations, and the business model supported by the program [14].

Designers and developers are building some propositions about the developed proposed system and the risks that can be happened. The specialists in security and risk should help to challenge these propositions against generally accepted best practices. Risk analysis is not a science Because of this, knowledge and experience can’t be exaggerated, and comprehensive knowledge about defects, vulnerabilities and threats is most important to the success factor [15].

A proto-typical analysis can be done by several major activities, which often contain many basic sub-steps:
- Learning the analysis objective, consist of: Reading and understanding architecture documents and specifications. Studying the code and another software artifact.
- Discussing the security issues that surround the software, consist of: Discussing about the product working way. Define disagreement areas.
- Determination of the compromise probability.
- Rank risks.
- Develop a strategy of mitigation, Include: A sub step to mitigate risks is recommended countermeasures.
- Report findings, contain: Describing minor and major risks carefully. Give high attention to the impact of risks.

6. Risk Analysis and Requirements
Risk analysis started at design requirements stage, and must consider that the risks developer trying to count risk. Let us look at three methods to interpolate the philosophy of risk-based into the requirements phase (note that the systems requirements depend on UML gave more attention on security functional requirement than they do on abuse and bad use cases) [16].

Secure UML is a methodology for modeling the policies of access-control and the integration into software development derived model. Secure UML is depend on models security requirements and role-based access control for well-behavior applications in specific environments and expected situations. UML-sec is an extended UML that provide the ability to perform modeling of security-related features as trust, confidant and access control. Guttorrn Sindre and Andreas Opdahl [15] try to model abuse cases as a method to understand how applications may respond to threats in a minimum control environment; they make a good specification of some functions that the system should prevent. Many degrees or levels of risk probability that Shawn in table 1.

| Risk probability | Description |
|------------------|-------------|
| <10%             | very low    |
| 10-25%           | low         |
| 25-50%           | moderate    |
| 50-75%           | high        |
| >75%             | very high   |

7. Conclusion
We conclude that the developer must give attention to each identified risk and make an assessment to the seriousness and probability of it during the risk analysis process. There is no simple approach to do this; developer must depend on his judgment and experience, which is the reason of the suitable person
to help with risk management are experienced project managers. Most of the risk estimations should be based around a number of ranges and not be a precise numeric assessments. The probability of the risk can be classified as shown in table 1. The side effects of these risks can be judged as insignificant, tolerable, serious or catastrophic, or. In general, catastrophic risks should be gave high attention and considered carefully, as should all other serious risks, which have probability of occurred more than a moderate level. From this review study, we found that there is no reliable published dataset contain practical features of software projects development, so it is necessary to create like this dataset to support software developers in risk analysis and risk prediction by using Data mining technique.

References
[1] Noor Habibah Arshad, Azlinah Mohamed, Zaiha Mat Nor, “Risk Factors in Software Development Projects”, Proceedings of the 6th WSEAS Int. Conf. On Software Engineering, Parallel and Distributed Systems, Corfu Island, Greece, February 16-19, 2007, pp 51-56.
[2] Paul L. Bannerman, “Risk and risk management in software projects: A reassessment”, The Journal of Systems and Software, ScienceDirect, Elsevier, pp 2118–2133, Available online 23 April 2008.
[3] Robbie T. Nakatsu, Charalambos L. Iacovou, “A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: A two-panel Delphi study”, Information & Management, ScienceDirect Elsevier, pp 57–68, Available online 10 January 2009.
[4] Ozren Đurković, and Lazar Raković, “Risks in Information Systems Development Projects”, Management Information Systems, Vol. 4 (2009), No. 1, pp. 013-019.
[5] Srikrishnan Sundararajan, Bhasi M., Pramod K.V. “An Empirical Study of Industry Practice in Software Development Risk Management”, International Journal of Scientific and Research Publications, Volume 3, Issue 6, ISSN 2250-3153, June 2013.
[6] Haneen Hijazi, Shihadeh Alqrainy, Hasan Muaidi, and Thor Khdour, “Risk Factors in Software Development Phases”, European Scientific Journal, January 2014 edition, Vol. 10, No.3 ISSN: 1857 – 7881 (Print) e - ISSN 1857- 7431.
[7] Ewa Ziemba and Iwona Kolasa, “Risk factors framework for information systems projects in public organizations – Insight from Poland” Proceedings of the Federated Conference on Computer Science and Information Systems pp. 1575–1583, DOI: 10.15439/2015F110 ACSIS, Vol. 5, 2015, IEEE.
[8] Maruf Pasha, Ghazia Qaiser, And Urooj Pasha, “A Critical Analysis of Software Risk Management Techniques in Large Scale Systems”, IEEE Access, Special Section On Software Standards And Their Impact In Reducing Software Failures, PP 12412- 12424, Digital Object Identifier 10.1109/ACCESS.2018.2805862.
[9] Ungureanu Anca, Braicu Cezar, Ungureanu Adrian, “Risk Identification in Project Management”, Procedia of Economics and Business Administration, ISSN: 2392-8174, ISSN-L: 2392-8166, Available online at www.icesba.eu, 2015.
[10] G. McGraw, “Software Security,” IEEE Security & Privacy, vol. 2, no. 2, 2004, pp. 80–83.
[11] H. Cavusoglu, B. Mishra, and S. Raghunathan, The Effect of Internet Security Breach Announcements on Market Value of Breached Firms and Internet Security Developers, tech. report, Univ. of Texas at Dallas, School of Management, Feb. 2002; www.ut dallas. edu/~huseyin/breach.pdf.
[12] M. Howard and D. LaBlanc, Writing Secure Code, 2nd ed., Microsoft Press, 2003.
[13] G. Hoglund and G. McGraw, Exploiting Software, Addison-Wesley,
[14] J. Viega and G. McGraw, Building Secure Software: How to Avoid Security Problems the Right Way, Addison-Wesley,
[15] G. Sindre and A.L. Opdahl, “Eliciting Security Requirements by Misuse Cases,” Proc. 37th Technology of Object-Oriented Languages and Systems (TOOLS-37), IEEE CS Press,
[16] Object-Oriented and Classical Software Engineering (Eighth Edition) Stephen R. Schach.