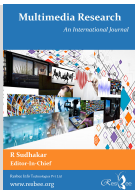
Vol.5 No.3 July 2024

Multimedia Research

Received 7 Apr, Revised 20 June, Accepted 27 July



Enhanced Software Risk Assessment in Software Development Lifecycle

Kuma Yadi
Ethiyopia

Abstract: Risk is a destructive and serious problem that may happen in the future of the software project. The definition of risk is an indicated negative thing. However, in software project development risk has two meanings. One is the opportunity for the developer and customer if it gives motivational consequence on the project objectives and the other one is a negative risk that affects the project objectives and may lead to the failure of the project. Improper assessment of negative risk is the main reason for the failure of SP.

Keywords: Risk, Software Risk, Software Risk Assessment, Software Risk Management Process

1. Introduction

1.1Background

Risk is a serious problem and probability of possible or impossible losses in the future. Software project risks are factors or hazards that do not affect the projects [22]. The main factor of these risks is project personnel, software complexity, product requirements, software availability and reliability, evaluation methods, monitoring techniques, development processes, and tools used. A situation that harms project objectives, especially scope, quality, time, and cost is software project risk [46]. In this case, software risk is a situation that may put pressure on the project results. However, without considering the outcome, it is essential to recognize risks, assess the likelihood of events, and assess their effectiveness. Every participant in project development is a software engineer; managers and stakeholders must participate in risk analysis and management. Software project risk is increasing in the software industry. Due to poor time management, budget, performance, user, and functional requirements, applications, design architecture, and unproven technology [44], some software development projects have failed to achieve the goal of delivering valuable software projects.

A risk is a destructive event that may occur during SDLC with difficult consequences [6] this risk may happen because of unclear requirements, lack of user participation, lack of experts, contract failure, and technological progress. Software projects cannot completely get free of risks; however, by incorporating appropriate risk management techniques, the possibility and risk impact can be minimized. Assessing risk is a basic activity in the software project management process [23]. Since the expectations of developing a project are uncertain, its success depends on the correctness of the ongoing assessment. Risk analysis calculates the probability (desirable or undesirable) of occurrence and its impact [4]. There are many published risk assessment models for software projects. It is expected that these models meet the needs of managing software risks in software projects.

The software risk assessment framework is a method of prioritizing and sharing information about classified risks faced by software development organizations [13]. The software risk assessment framework determines the general classification based on risk and determines the priority of the process. These frameworks are proposed for two phases of large-scale development of software the elicitation phase and the deployment phase. Assessment and evaluation of uncertain events through the SDLC are very useful for software project management [32]. Existing methods have measured the number of software risk factors. However, these technologies use risk factors to evaluate software risks, and there are some disadvantages, such as expensive processing costs, complex calculations, and the selection of

https://doi.org/10.46253/j.mr.v5i3.a5

technical risks that can improve the performance of evaluation and assessment. Without software scale, small, medium, and large software projects, the risk may affect the quality of major areas [5]. SRA identify source of software projects risk in the organization, level them depend on their importance, and determine the activity manager for software project execution to manage the identified software project risk.

1.2 Problem Statement

Software development uses different technological achievements and requires conceptual understanding. Therefore, all software development projects(Jiwat, 2019) contain elements of ambiguity. Today, software development projects still cannot convey adequate systems in terms of budget and time [20]. In a strategic and important environment, there is no management as a project, so projects must be executed parallel to achieve organizational goals. One of the basic goals of any software industry is to create software that satisfies customers' needs. Because of the nature of software projects, risk events have become one of the focuses of the industry. Therefore, it is essential to ensure that the development procedure has a feasible risk assessment procedure. The current improvement of Software Engineering is a fast-growing industry [26].

Proper Risk assessment increases the opportunity for software project improvement [6]. Major Software risk is a factor that challenges project performance and even leads to failure. Improper Software risk assessment is the main intention of the failure of a project because the entire software project risk analysis is carried out during the project development procedure. Even with a lot of research and development in the field of software project development, software projects still cannot deliver quality products within the estimated budget and estimated time [18]. Software projects cannot completely get free of risks; however, by adopting appropriate risk assessment techniques, the possibility and damage can be [19]. Even with advancements in technology and development processes, software projects continue to face serious problems repeatedly.

Researchers suggested different techniques for risk assessment that could be applied to resolve certain risks. Risk identification, risk analysis, and risk classification is reported [22]. Researchers investigated risk identification (Zaidi & Pelling, 2014), risk analysis, and prioritization [4] for risk assessment. [13] suggested that only elicitation and deployment phase software assessment techniques such as risk identification, risk analysis, risk evaluation, and prioritization. Also [39], risk assessment processes such as risk identification, risk measurement, and risk evaluation are being developed. However, still, software projects are not free of risk because the frameworks are not in detail. Once they assess the risk they didn't review it, this may lead to another risk due to requirements change, technological improvement, tools and techniques, and Human resources. Furthermore, here the technical risk is the limitation. However, the enhancement of risk assessment is needed to solve such changes during the development process by reviewing the assessment and re-assessment if necessary. This includes documenting risk, communication between customer and team, and revising the steps of assessing the risk at the final stage. Furthermore, this identifies source software project risk at each phase of SDLC.

- ✓ Which factors are important for defining software project risk issues?
- ✓ How to assess software Project risk using the predicted probability that hinders the objective of a software project?
- ✓ Which machine learning algorithms SVM, logistic regression, and Random forest are more accurate for predicting software project risk?
- ✓ How to evaluate the performance of the proposed model?

1.3 Objectives

1.3.1 General Objective

The general objective of this research is to develop a model for enhanced risk assessment in the software development lifecycle.

1.3.2 Specific Objectives

To achieve the above-mentioned general objective, this research addresses the following specific objectives.

- ✓ To determine the software risk issue that affects the quality of the software projects.
- ✓ To rank, assess, and evaluate risk factors based on their importance, and frequency of occurrence
- ✓ Calculate predicted probability to determine software risks

2. Literature Review

2.1Software Project Risk

Risk is a desire for failure, and a harmful problem that could or could not happen in the future. Risk is mainly caused by the need for data, control, or time. The possibility of being troubled by losses in the software project development process is a software risk. Loss refers to the increase in production costs and the decrease in product quality; all these factors can resolve the completion of the project on time. Software risks can be either internal risks or external risks. Internal risks are controlled by the supervision of the project manager, while external risks are outside the manager's capacity. Software project risk refers to uncertain events that, if they occur, damage the project goals [40]. Reasons can be assumptions, requirements, and constraints that may lead to impact or opportunity. The dictionary meaning of risk is negative meanings, however, when seeing risk in a project, it has both positive and negative aspects; which lead to software projects in terms of scope, schedule, cost, and quality none of them reached their goals. This is the reason behind progress and effort; however, there is another risk, which is the risk of downtime without proper assessment and control.

2.2 Software Risk Management (SRM)

SRM is used to reduce, control, and manage the steps of risk. In a software project, RM is a permanent process of identifying, analyzing, evaluating, monitoring risks, and mitigating the adverse effects of losses. Generally, it is accountable for the identification, analysis (or measurement), processing, and monitoring of risks. The core function of project SRM is to identify any related risks and take preventive and corrective measures to prevent or minimize negative effects. Depending on the particular department, project risks may evolve in many different forms. SRM (sometimes called "uncertainty management") is a systematic process followed by a company to minimize the possibility of unexpected events and maximize profits. It refers to increasing the possibility and effect of opportunities and reducing the possibility and effect of project objectives [50]. In this process, the decision accepts known or assessed risks and/or takes actions to minimize the consequences of risk [42].

Others define SRM as minimizing the occurrence of negative risks, thus avoiding the loss of risks organization. Therefore, SRM is a process whose main objective is identifying the risks and opportunities faced by the project or business in its early stages, and taking actions according to the necessary response strategies to mitigate or use the risks to ensure the success of the business. Because risks and opportunities are uncertain, SRM can also be called uncertainty management [42]. Although there are many methods in SRM, the risk of failure in software project development is high. Therefore, if the complexity and scale of software projects are increased, it becomes more difficult to manage software development risks. Also, the optimization method has been tested through various software project risk prediction models. Fig. 1 briefly describes the SRM phase of a successful and complete software project.

Fig.1 *Software Risk Management Process*

SRM is a complex activity and key activity in a software project [33]. SRM methods proposed so far mainly follow two characteristics-probability and impact. Most of the proposed methods are static and can perform qualitative and quantitative analysis to assess and control risks. Most methods divide SRM into some basic processes, such as identification risk, analysis, risk monitoring, and control [43]. In the planning stage, SRM activities are mainly focused on risk assessment, which is a discovery process used to identify potential risks, analyze or evaluate their risk impact and decide the level of risks. The focus of software risk identification activities is to enumerate possible risks, create risk statements, and

determine possible risk conditions as deliverables. Based on the risk statement and risk background, various aspects of uncertainty can be evaluated and prioritized so that project managers can determine where measures should be taken to manage such risks [31]. Therefore, the RM stage provides information about the number of risks and an estimate of the correlation between each risk.

SDP is not yet certain[24]; SDLC phases are vulnerable to risk. These development stages are susceptible to various risk factors, which can hinder the completion of the project. Control of these factors requires an understanding of the problems and uncertainties of the basic software project development process. Therefore, risks are identified first to recognize risks that threaten the accomplishment of software projects. These risks may prevent the project from achieving its expected results, and this may cause the entire project to fail. The goal of SRM is to minimize the effect of risks. RM uses technology and planning to achieve its goals. RM benefits are direct (primary) benefits related to major risks, products, personnel, and costs. Indirect (secondary) benefits related to process improvements, such as optimization, pragmatic decision-making, better process management, and alternative methods. RM is to prevent and control risks before they get worse, so, risk monitoring and maintenance steps are adopted in the RM plan [29]

SRM involves thoroughly ing the system or process to identify problems or potential risks, analyze them, and develop strategies to mitigate and control risks. Mitigating software risks does not mean immediately eliminating all risks that caused the failure. However, the risk should be reduced to an appropriate stage. During risk identification, understanding all major risks, their severity, and potential losses. Risk mitigation or control can be based on knowledge of the overall risk. This strategy helps software developers complete the project within time and budget. Projects with effective RM, in addition to reducing costs and time, also tend to produce better-quality output. The SRM steps assess the risks in the phases of development [48]. RM is the entire process used by project managers to reduce and control risks.RM phases are risk identification, assessment, and risk response control. SRM is related to risk quantification and is interpreted as:

- ✓ Briefly describe the risks faced in the project
- ✓ Specify the occurrence and cause of the specified risk probability
- ✓ characterize the effect of the risk on a specific project
- ✓ Define the loss of potential risk

2.2.1 Software Risk Management Process

According to [30], RM is a systematic procedure for planning, identifying, analyzing, monitoring, and controlling project risks. It involves tools, processes, techniques, and steps. These tools, processes, and techniques need to minimize the possibility and consequences of negative risks and increase the chance of project goals. RM practices involve the systematic purpose of processes, management strategies, and procedures to identify, analyze, evaluate, monitor, communicate, and project RM is effective throughout the development phases. The SRM process includes the following processes:

Risk identification can identify a project, product, and business risks
Risk analysis assesses the likelihood and consequences of these risks
Risk plan Develop a strategy to reduce the effect of these risks
Risk monitoring can monitor the risks in the entire project
Fig.2 briefly describes the SRMP through the development cycle

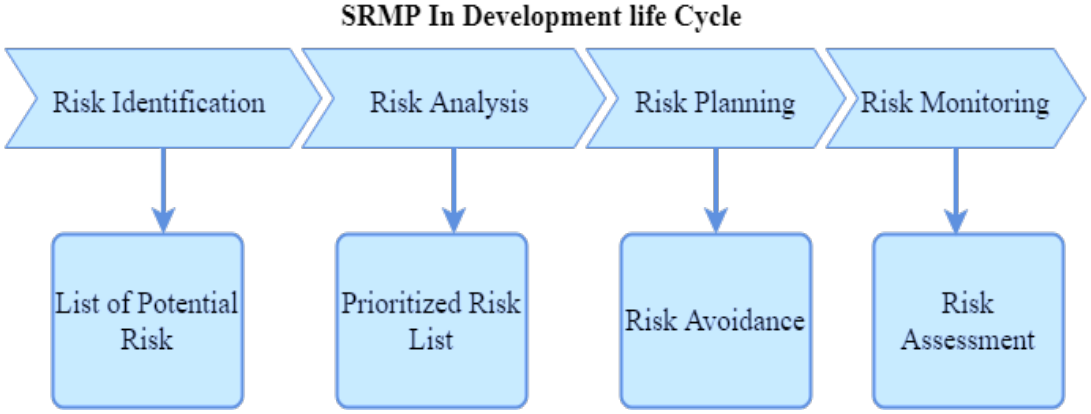


Fig.2. *Software Risk Management Process in SDLC*

2.3 Software Risk Categories

As [29] distinguished, software risks are mainly divided into three categories: technical risk, schedule/scope, and cost. Technical risks are associated with the performance of software products, including functional, quality, reliability, and timeliness issues [29]. However, if there is no change in the project requirements, scope, design, and implementation phases, unexpected technical risks may occur and are the source of the project failure. To minimize such failures, the PM must understand the currently used technologies, techniques, and methods needed to mitigate risks. Even if there are many more experienced technicians, there are still unexpected technical risks today. The schedule and scope risk is associated with the schedule and scope of the software product during development [29] It is possible to make scope changes in an SP to some extent; however, without detailed specifications, some problems may occur after the implementation process begins. These basic changes often require change requests that cause progress issues.

When other components rely on this function, it may cause a system failure. If a complex project is developed by a low-tech developer, schedule risk may occur [9]. When developers develop complex projects within a tight schedule, the schedule risk is even more uncertain. Software cost risk is the estimated cost of the software product during the development stage and the final delivery period, which may include budget, non-recurring costs, recurring costs, fixed costs, and variable costs [29] ability and impact. The team combines these two dimensions and multiplies them by them to calculate the risk score, so the risks can be sorted easily. Product risk is a software project risk associated with a software product as a deliverable product [9], which is affected by the following product-related parameters: reliability required by the software, product size, complexity, and database and documentation requirements. According to [9], personnel risk is the main source of project risk and affects the overall productivity of software projects. Human risk is related to the overall ability and experience of the analyst, and the specific experience of the programming language and tools.

In general, there are several risk categories however, they didn't have an equal impact on software project development. Most researchers studied risk categories such as budget risk, project risk, business risk, and technical risk. The probability and its effect vary depending on the project domain area and tools and techniques used in the project assessment. Software projects fulfill the two user requirements i.e. functional and non-functional requirements. These requirements are gathered and analyzed in the SDLC. During these cycles there are several risks may occur. Therefore this focuses on software project technical risk in the development phase

2.4 Software Risks Assessment (SRA)

SRA is an innovative process that can identify risk sources, analyze or evaluate potential risk impacts, and prioritize risks. SRA is a method of identifying risks by identifying opportunities and impacts, analyzing risks, and determining risk priorities. After the measurement, the level of effect is divided into highest risk, high risk, medium risk, or low risk according to the chance and risk impact that occurred risk [12]. SRA is the method of calculating the probability (desirable or undesirable) and its impact. This step can help select less risky projects and eliminate residual risks. First, use a risk identification tool to identify major threats and positive events that may affect the process and quality of the product. After determining the main risks, the second stage conducts an appropriate assessment of the frequency and results and then ranks the various risks according to their risk score results. In this case, by comparing the identified risks and the next step of the RM process, an appropriate risk response strategy can be determined [3].

SRA is the main activity in the project planning stage and is essential for determining the realization of the software project. However, in terms of popular RM methods, the realization of RA activities is highly dependent on human judgment and experience [10]. Since the future of the project is uncertain, its success depends on assessing risks in advance. During the risk identification process, the potential risks of the SPs are determined. After identifying various risks, the severity of their potential impact should be assessed. In the risk mitigation activities, an effective risk reduction plan has been developed to minimize the damage of encountering risks. After completing the risk management plan, the next step is to monitor risks. The risk monitoring process reviews planned activities and modify them. The purpose of SRA is to recognize new risks as much as possible within the allowed time frame, and how to take corrective measures to mitigate various risks [23]. SRA is the overall activity of risk identification, analysis, and evaluation. Identifying risks includes understanding the risk, the scope of its impact, the event, and possible measurement. The goal is to create a complete list of risks, including risks that may be related to missing opportunities and risks that are outside the organization's direct control [12].

Risk identification is the process of judging and classifying current and/or potential risk sources or risk factors and identifying risk attributes. Brainstorming is a popular applied method of risk

identification [36]. These factors hinder the project from accomplishing its objectives, leading to project failure. Identifying software risk factors is a key activity, and the success of RM depends on identifying the main risks that affect the project during project development [24]. Although all the ideas or issues that arise could or could not be relevant, it is significant to document all issues (risks), possible effects, and accepted solutions [13]. Identifying risks is not part of a single area or phase of software development; however, a necessary activity involved in the entire project. This activity involves all stakeholders affected by the risk, such as RM experts, project managers, team members, and customers.

Risk identification starts at the point of project conception and ends at the point of project deployment because risks can appear at any time and development phase [41]. The initial process of risk assessment, namely risk identification, has completed all identifications of risks in software projects. Work products, processes, plan reviews, meetings and brainstorming, investigations, and lists of all possible risks are included in methods that help identify risks. Risk evaluation always goes hand in hand with risk. Risk evaluation computes the likelihood and impact of due to risk. Risk analysis includes the identification, analysis, planning, monitoring, and resolution of risks [34]. To analyze the impact of risks involved in software development, project managers must identify risk sources. Software risk components are performance risk, support risk, and cost and plan risk. In this step, determine the acceptable degree of risk; the determined risk may be acceptable or unacceptable.

At-risk analysis, the identified risks are rated according to their probability of occurrence. Then, after defining risk levels to set priorities and control them, calculations are based on existing assets or investments in new resources. The qualitative techniques of risk analysis can better determine the behavior of the main risks, and once analyzed using a risk matrix, different risks can be shown [13]. Risk prioritizes [13] sorts and organizes uncertainty as the degree of an event on the project and organizational resources. It can be avoided, reduced, communicated, and ordered. Corrective actions are in use to remove all potential risks (avoid situations).

Several researchers have studied the RA steps and follow-up. Most of them stated risk identification, evaluation, and prioritization. Others also described software risk assessment steps as risk identification, analysis, evaluation, and prioritization. However, the appropriate software risk assessment needs additional steps to produce a quality product at the closing stage. Thus this enhances the software risk assessment stages to risk identification, risk categorization, risk analysis, risk evaluation, risk prioritize, risk mitigation, risk document, and communication and review of the assessment and reassessment if necessary. Maybe this assessment is applied to the complex project as the nature of the project increases many risks arise at each phase and stage. To correctly analyze these risks it is also essential to assess them in detail and record their effect as much as possible. During development, the main source of risks is a requirement. These requirements are gathered from customers. To avoid such kind of risk the customer and development team have clear communication unless the risk may cause to fail the project. Maybe during development, there may be a change in requirements, implementation tools, project manager, delivery date, and budget may be changed due to some reason. Therefore reviewing the assessment and modifying it if a reassessment is needed is possible.

3. Methodology

3.1 Software Risk Assessment (SRA)

SRA is the central idea of software project RM. The two main components of the SRA process are qualitative risk assessment and quantitative risk assessment. Qualitative risk assessment deals with the qualifying risk events, while quantitative risk assessment studies the damage and possibility of the recognized risk events. The central concept of uncertainty is that it is an event that could or could not happen, so its occurrence is uncertain and brings adverse economic effects to software projects. The risk impact is a random quantity, so it is modeled using random variables with potential probability distributions. Unless a risk event occurs, the actual risk impact cannot be achieved; the inability to fully understand the risk event and its impact before it occurs is the uncertainty surrounding the risk event. The traditional risk assessment model can be used as the basis of risk assessment in the project development stage.

Risk assessment can identify events that may harm the software development organization. These impacts include potential risks; the time required to succeed in the project or the time to resume operations, and the time to manage ways to properly mitigate the occurrence of uncertain events. Risk assessment supports determining which steps (if implemented correctly) can reduce the damage of the incident. The broader risk assessment process usually includes:

- ✓ Identify the problem that caused the risk,

- ✓ Analyze their significance,
- ✓ Identify appropriate mitigation techniques such as accept, avoid and transfer
- ✓ Determine which mitigation strategy is most appropriate, and
- ✓ Communicate results and suggestions to decision-makers.

The importance of risk assessment varies across the industry. In general, the main goal is to support organizations in assessing and managing risks. The objectives include the following:

- ✓ Provide possible risk analysis
- ✓ Prevent injury or illness
- ✓ Meet legal requirements
- ✓ Establish risk awareness
- ✓ Establish an accurate evaluation method
- ✓ Justifying the cost of managing risk
- ✓ Determine the budget for remedial risks

The SRA phase proposed in this includes software risk identification, classification, analysis, prioritization, software risk response, documentation, and review, evaluation, and modification as required.

3.1.1 Software Risk Identification

SRI is a complex process that recognizes the risks that may be faced in the development phases. It is first necessary to identify the problems that occurred in the previous project. By converting the project plan into a flowchart and identifying the necessary areas, the project plan can be properly analyzed. In the risk identification stage, it is essential to apply appropriate software risk identification tools. It is promising to correctly evaluate the decisions made on technical, operational, organizational, budget, time, functional, and non-functional factors. Fig.3 shows how to determine the risks based on past projects, current projects, and future projects

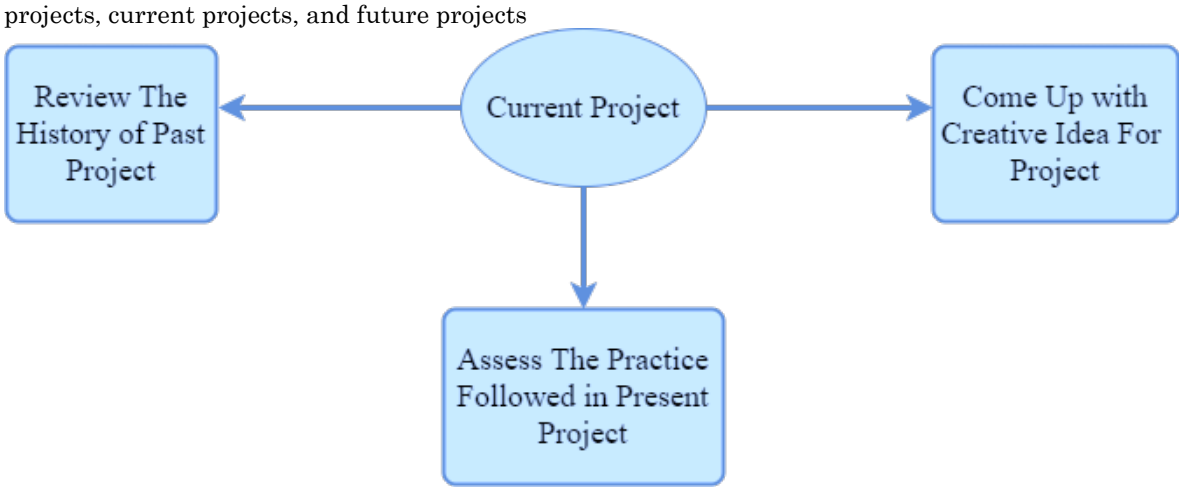


Fig.3. *Component of Risk Identification*

3.1.1.1 Risk issue in SDLC

Risks in software projects can be classified as risks that occur during the SDLC phase, namely requirements, analysis, design, coding, testing, and maintenance. Besides, there are other sources of software project risk, such as plan, quality, time, and cost. Common software risk factors related to software project development [24].

- ✓ Inaccurate resource estimates
- ✓ Customers are uncertain
- ✓ unclear requirements
- ✓ Risk of design errors
- ✓ Insufficient management process
- ✓ Tools, technology, and working environment

The technical risks are very complex and can lead to product functional and non-functional failures [43]. The School of Software Engineering determines the technical risks associated with the technical achievement project as follows.

- ✓ Lack of popular framework
- ✓ Lack of capability to familiarize with technological progress
- ✓ Developer issues

- ✓ poor project management
- ✓ Too much confidence in the ability of technology to solve problems

The software development phase is the risk phase [24]; from the beginning to the end of development, these phases are all vulnerable to risk. Risks from the software development stage hinder the achievement of the project. Software risk factors are possible circumstances and harmful effects that affect the time, cost, and quality of the project. The main technical risk factors caused by improper use of hardware/software technology and techniques and software engineering theoretical principles are the cases for the failure of software projects [24]. There are also environmental factors related to environmental software. Management factors involve the management and control of time, human resources, and budget. Organizational factors are factors generated by the organization in which the software is located.

a)Risk issue at Feasibility

Feasibility activities are very important in the life cycle of software development. Especially in large projects, it is regarded as a stage before requirements gathering and analysis. It can decide whether to develop a software system. This phase helps to determine the risk factors faced in the development and planning risk analysis process. Table 1 lists and explains the risk issue in the feasibility stage

Table 1. Risk Issue at the Feasibility Phase

Risk factor	Description
Unrealistic Schedule	The estimated time for the entire project may exceed the previously agreed delivery date. To solve these situations, PM exceeded the developers and increased the quality requirements on time, thus unrealistically delivering on time.
Unrealistic Budget	The estimated budget for the project regarding required resources, time, and effort. The estimated cost for the project may exceed the available budget, and if it is not successfully calculated, the project may exceed the funds in the early stages and may fail.
Unclear Project Scope	Dealing with the goals, scale, and requirements of the project scope is the most basic task of a successful project manager. It is difficult to manage the degree of the project to determine what the project should perform because the project failed to achieve its goals and required additional activities. This is a big problem that PM is difficult to solve and may lead to project failure.
lack of resources	Resources such as tools, people, and technology are sometimes insufficient to accomplish the project. In other words; if the system requires new technology, the project cannot be implemented using existing technology. However, if the project under development requires new technology, the developer uses the existing old technology, which may threaten the successful implementation of the project, thereby exposing the developer to the risk of technical changes.

b)Risk Issue at RequirementsGathering

Many risk factors appear in the requirements collection stage, and application areas need to be analyzed, functional requirements and non-functional requirements are collected, reviewed, and clarified. Table 2 describes the risk factors when the requirement is gathered.

Table 2. Risk Issue at the Requirement Gathering

Risk factor	Description
ambiguous Requirements	If developers and system analysts do not understand the requirements, the requirements become blurred.
Incomplete Requirements	When the requirement lacks certain user requirements, constraints, and other quality requirements, it means that the requirements are incomplete. As all know, users cannot describe requirements at the start of the project. Therefore, the requirements continue to change during the development phase.
Inaccurate Requirements	When the requirements do not meet the actual user needs. It indicates that the requirement is incorrect.
Lack of non-functional requirements	Usually, developers and system analysts focus on functionality and ignore non-functional requirements. Non-functional requirements are critical to the success of the project.

contradictory user requirements	If the system has different users, they have different needs, not only different needs but also conflicts [24]. Therefore, this leads to conflicts when analyzing needs.
Unclear explanationof development environment	In many cases, system analysts cannot clearly understand the real world in which the software runs.

c) Risk Issue at Requirements Analysis

After collecting the requirements, the software analyst analyzes the requirements. At this stage, requirements are analyzed, classified, organized, and prioritized [24].

Table 3. Risk Issue at Requirements Analysis

Risk factor	Description
Non-verifiable Requirements	Without limited cost-effective processes, such as testing, inspection, demonstration, or analysis, the requirements cannot be verified [24].
Infeasible Requirements	Without sufficient accessible resources for implementation, these requirements become infeasible. To put it another way; if the quality requirements are not correctly implemented on the project, then this is not feasible.
Inconsistent Requirements	If the requirements contradict any other requirements of the system, inconsistent requirements occur [24].
Non-traceable Requirements	The source of each collected requirement must be understood to facilitate future documentation and reference.
Unrealistic Requirements	This shows when the requirements are unclear, inconsistent, incomplete, unfeasible, and unverifiable. Therefore this is an unrealistic requirement.

d) Risk Issue in the Design Phase

After analyzing the requirements, the next stage is the design stage. This phase is used to create the whole system[24]. At this stage, the software system is replaced by the main components and the relationships between them. There are many activities, such as requirements documentation, architecture design methods, programming languages, and verification, specification, and document design activities. Table 4 illustrates the risk issue in the design phase.

Table 4. Risk Issues at the Design Phase

Risk factor	Description
RD isn't clear for developers	When developers are not involved in the requirements initiation and analysis phase, the requirements document may not be understood by them. Therefore, they are not able to develop designs based on the basic knowledge required by the system, so they may develop designs for systems other than the expected system.
Improper AD method Choice	These may be the reasons why there is no standard building design. If the developer chooses the wrong AD method, the implementation may not be completed successfully, and the AD may not work properly. When choosing AD, one must consider the PL, because it has its impact on AD.
Improper choice of the PL	The wrong choice of PL affects development activities in many ways. It does not support AD. Improper selection of PL reduces the simplicity and modification of the system.
Complicated Design	If developers do not have enough experience and skills to manage the complexity of this system, then they develop more complex designs.
Large size components	Large components are also decomposed into other components. These components may also be complex to implement, and it is hard to determine the core functions of the components, so challenging to assign functions to these components.
Unavailable expertise for reusability	In software development, reusability is not forever the right choice. If there is no available expertise to maintain old components for reuse, this method can be used. This is a risk because it may hinder the project and delay its progress. In this case, the team often develops components from scratch, which may also delay the progress of the project.

Less reusable components than estimated	If the estimation of the reusable component is incorrect during the analysis phase, the component must be developed from scratch. This may affect the estimated time and budget required to succeed in the project. Developers are surprised that much of the ready and reusable code must be rewritten from scratch, which causes project delays and budget overruns.

e) Risk Issue in the Coding Phase

After designing the entire system model, the next step is coding. In the coding phase, the developed design model becomes coded in a programming language. The source code module is tested in the unit test step. Table 5 illustrates the risk factors at the coding stage.

Table 5. 3Risk Issue at Coding Phase

Risk factor	Description
Non-readable Design Document	This happens when designing large documents. It may not be understood or understood by the developer, so it is complicated to decide what code to write.
Programmers cannot work independently	If the design documentation is incomplete, then programmers could not work independently because they must make their own decisions to fill in the gaps in the design documentation, which may affect programmers who develop other components.
Develop wrong user functions and attributes	The implementation of functions and attributes depends on the detailed list of design specifications in the design document. If the design documents are inconsistent, incomplete, and unreadable, programmers may develop the wrong functions and properties [24].
Develop the wrong user interface	It is essential to develop an attractive user interface; to clarify the usability and understandability of the system, thereby increasing user acceptance. Otherwise, the project may fail. A well-designed user interface requires a full acceptance of user desires and detailed specifications during the design phase.
PL does not support architectural design	If the selected PL does not support AD, then the programmer falls into the problem of not being able to use the selected PL to implement AD.
Modules are developed by different programmers	In large projects, the development team usually has many programmers. These programmers work on different components and follow creative and coding methods, which lead to complex, ambiguous, and inconsistent code. Also, if they are developed on the same component, it leads to various forms of the similar component.
Complex, ambiguous, inconsistent code	If programmers do not apply coding principles and practice programming; this can lead to complex, large, and ambiguous code.
Different sort of The similar components	If many programmers develop the same component, there may be diverse sorts of the same component, causing integration problems.
Developcomponents Start from scratch	If there is no available expertise to maintain existing components for use in current systems, developers tend to develop components from scratch. This may take time and requires more energy.
A large amount of duplicate code	This can happen when writing certain code segments repeatedly. I do it manually, it needs time, effort, and budget.
Inexperienced Programmers	If the programmer does not have enough experience in the selected programming language, many syntax errors may occur, which can lead to complex code, ambiguity, functional errors, and possible user interface development.
Too many syntax errors	If the selected PL is sensitive and the excellence of the compiler and debugger is poor, this can cause syntax errors when writing code, especially when the programmer has no experience in the selected programming language.
Technological change	The project may involve applying new technologies that have never been used before. During this time, programmers/developers may discover the complexity to deal with these technologies.

Table 6. Risk Issue at the Testing Phase

Risk factor	Description
Newly designed component high fault	When the components are built from scratch after it is used and tested for the first time, this indicates there are undiscovered errors and many faults might be found.
The code is not understandable by reviewers	When the testers are doing testing, they have to review the code from time to time to debug the errors that caused the component's faults. If this code was not clear they would not do that.
Lack of appropriate computerized testing tools	However, if the testing activity is poorly automated, it is boring and monotonous. Testing is a large discipline in the software development phase, few tools are available that support testing. Most of the currently available testing tools are used for coverage analysis.
Testing is monotonous, tedious and repetitive	As stated before, if the testing were not automated, it is monotonous and boring and leads to failure to produce products.
Informal and poorly-understood testing procedure	Frequently, the testing process is an informal activity by adapting intuitive techniques because it is considered a complementary (not essential activity), and short-term training is given to the developers on testing activity.
Poor test cases documentation	Test cases were documented automatically while doing testing for Test cases have to be documented automatically while doing testing for valuable future use for similar cases.
Not all faults are discovered in unit testing	Some errors remain hidden during unit testing; this is the reason testing techniques are currently in use and lack automation testing.
Poor Regression Testing	RT is for modification or reruns of all the already successful run-affected test cases when the modification is made to the currently available code. It also saves budget and time.

g) Risk Issues at the business and Modificationstage

During the development of software projects, the last stage of this development phase is the maintenance phase. These phases are the phase in that the software is delivered to the customer and deploys; the user acceptance test and the measure of risk exist. This phase includes the following activities: installation, operation, acceptance testing, and maintenance [24].

Table 7. Risk Issues in the Operation and Maintenance Phase

Risk factor	Description
Problems during installation	If the deployers haven't enough skills in the system's nature and how it works, particularly the complex nature of the system and challenging environment, it is hard to install the system or install it incorrectly.
The installation problem in a real environment	Installing the system may affect the environment it works in. frequently, the users don't accept the action. If this occurs, it is insignificant.
Change in environment	During installing the system, deployers were surprised that the system can't be deployed correctly due to the changing environment, i.e. hardware improvement. This environment change is inevitable if it lasts a long time from system analysis to delivery and installation.

The appearance of the new requirement	During operating the system, users find that new requirements have to be implemented to fit the current actual user business, needs, and environmental and organizational changes.
System difficult for users	It is known that customers discover that it is hard to use newly installed systems. However, if this lasts along, it might affect the acceptability of the system.
Less description of the problem by customers	After the problem is reported by customers, Software engineers try to understand the problem by asking end-users questions that cause the problem.
Problems in Maintainability	Sometimes, it is hard to change the system due to the nature rigidity of its architecture or constraints forced by users or developers.

RI deals with specifying major project risks and ing their characteristics. It is the same as the risk register reason the result is out there. The collected data is used for risk analysis. This technique of risk assessment starts at the early phase of the project. Risks are identified and dealt with it as early when it is in starting the development phase. This process is not finished in one phase of development; it continues through the development lifecycle with main attention during key milestones. Some software risks are assessed by the development team, especially those who know the risks, however, others may be rigid to assess easily by the team, but predictable. The risk register is the medium of the recognized risk records through the development phase, which store in the central project. The risk identification inputs and appropriate techniques and tools which are used for risk identification and producing the output can briefly explain in table 8.

Table 8. Software Risk Identification Technique, Inputs, and Output

Inputs	Tools And Techniques	Outputs
➢ **Risk management plan** ➢ **Cost management plan** ➢ **Schedule management plan** ➢ **Quality management plan** ➢ **HRM plan** ➢ **Scope baseline** ➢ **Activity costs estimates** ➢ **Activity period estimates** ➢ **Stakeholder record** ➢ **project environmental factors**	➢ Reviews ➢ data gathering techniques ➢ Checklist analysis ➢ Assumption analysis ➢ Diagramming techniques ➢ Expert judgment	➢ Risk register

h) Techniques of Software Risk Identification

The following techniques and tools are appropriate to identify risks, which ensure that no significant potential risk is overlooked. Table 3.9 describes a brief description of risk identification techniques for producing quality products during the development cycle.

Table 9. Software Risk Identification Techniques

Tools and techniques	Description
Risk repository	The risk repository is the history data containing the record of recognized risks for completed projects. The risk repository is to arrive at a list of potential risks for the project. This risk repository can also be filtered based on risk sources, categories, and projects.
Checklist analysis	The risk identification checklist is a questionnaire that helps identify gaps and potential risks.
Expert judgment	Risk identification is also done by brainstorming or by interviewing professionals on project development and stakeholders.
Project status	The project status includes project status meeting reports, status reports, progress reports, and quality reports. This provides insight into the status of the project and potential new risks.
Documentation reviews	Documentation reviews are structured reviews (formal or informal) of the project documents example proposal/SOW (statement of the work), project scope, project management plan, constraints, estimates, budget, and schedule.

Information gathering techniques	This technique of risk identification includes brainstorming Delphi technique(this technique helps to reduce bias in data), interviewing, and SWOT analysis
Assumption analysis	Helps in identifying project risks and any changes to risk's impact
Diagramming technique	The cause of risks is identified by diagramming techniques. e.g. Network diagrams

3.1.2 Software Risk Classification

Software risks are internal and external risks. Internal risks are managed and controlled by the PM and external risks are above the management of the project manager. Two characteristics of risks are:

- ✓ Uncertainty-the risk could or could not occur.
- ✓ Loss-if a risk happens, useless outcome or damage occurs.

During risk analysis, the main goal is identifying the possibility of risk and the impact associated with each risk. To accomplish these, there are various categories of risks are considered. These are Project risks, Business risks, and Technical risks. Another categorization proposed by charette is known risks, Predictable risks, and Unpredictable risks

3.1.2.1 Project Risks

During project development project plan can be failed by project risk i.e. if the project risks become real, the project schedule may have problems and cost increase due to risk. The purpose of project risk is to identify schedule, budget, personnel (staffing and organization), resource, stakeholders, and requirement problem that have an impact on a software project. Project complexity, size, and the level of structural damage are also defined as project (and estimation) risk factors.

3.1.2.2 Software Technical Risk

Technical risks affect the quality and system performance of the software product. If technical risks become true, the execution could be difficult or impossible. Implementation problems, potential design, verification, interface, and maintenance problems are technical risk factors identified under the technical risk. Besides, requirement ambiguity and technical insecurity are also risk factors. These kinds of risks are relevant to the function of the system and performance as the name tells these are technical risks. These risks are caused by continuous requirements changing, the nature of the project implementation, and module integration.

In technical risks, we have to consider the following questions.

- ✓ Have the pre-project activities found the project to be viable?
- ✓ Are the technologies and tools well understood?
- ✓ Are the required environments available? Are they sufficient for each requirement?
- ✓ Are the project deliverables reviewed, accepted, and used?

For most software projects development, the main risk impact areas:

- ✓ Latestunproven technologies
- ✓ User requirements and functional service
- ✓ System Application and architecture
- ✓ System Performance
- ✓ Organizational

a) latest Unproven Technologies

Most software projects involve the use of improved technologies. Technology risks arise due to ever-changing tools, protocols, techniques, and standards. The technology risks occurrence probability increase due to the above reasons. Necessary training and enough knowledge of new technology by professionals are needed. However, improper use of improved technology leads to project failure.

b) User Requirements and Functional Service

Software requirements contain the user and functional service of the system. These requirements gather all user needs with general system features, constraints, functions, and quality metrics. Furthermore, the process of gathering requirements is tedious, complex, and complicated. Moreover, these requirements may change due to elicitation, prototyping, and integration activities. The change in one requirement may have an impact on the entire project, and modification of this requirement has an impact on other project factors. This conflict is due to requirements that lead to a more critical failure of a poorly planned software development project.

c) System Application and Architecture

To develop software quality product proper design system architecture that fit user needs have to play a significant role. Improper use of platforms, components, and system architecture has risks to the

system. Like technological risks, the team must include professionals who have enough understanding of the system architecture and the ability to make appropriate design choices.

d) System Performance and Organization

Several risks are there during the development of a software project that has an impact on the efficiency of the system. It is essential to include performance issues in the risk management plan. Especially technical risks affect the performance of the project in different ways. To test the system's performance it needs to ensure benchmarks and the direction developed product. The organizational problem has many effects on the project results. This may reflect in the team, human resources, and project managers. Efficient project management is a plan for the appropriate execution of the project and balances the team's needs and user expectations. Choosing team members who have good skills played a significant role in the completion of projects in the expected time, with expected quality, and fast to market.

3.1.2.3 Business Risks

Business risks have an impact on the practicability of the structure to be developed and put at risk the software project. Business risks are related to:

- ✓ Market risks that produce a product that cannot fit any customer requirements.
- ✓ Strategic risks create a product that is not related to the business plan of the company.
- ✓ Sales risks that salesforce has no understanding of how to sell
- ✓ Management risks area lack of senior management in controlling and giving attention to managing the business.
- ✓ Losing budgetary or personal commitment(budgetary risks)

3.1.2.4 Known Risks

Known risks are those that can be undiscovered after the careful analysis and assessment of the project risk, business plan, technical advancement, and reliable data source. For instance, unexpected delivery dates, lack of documented requirements or software scope, and poor development environment.

3.1.2.5 Predictable Risks and Unpredictable Risks

Predictable risks are extended from the long-ago project knowledge like poor communication, between team and customers, staff turnover, and estimated staff effort. They can and do occur, but they are very difficult to identify in progress. For each category, two types of risks have been categorized. Those are general risks and product risks. General risk is a potential threat to every software project. Only those who clearly understand the technology, personnel, and environment to be developed can determine product-specific risks. The project plan and the software scope statement were reviewed, and the answers to the following questions were asked to determine product-specific risks. What particular behavior of the result might threaten the project plan? The risk category provides a list of areas prone to risk events [14]. The organization recommends advanced standard categories, which must be expanded according to the project type. Most software projects have inherent risks because of a variety of potential problems. Experience in other software projects can assist managers to organize risks.

3.1.3 Software Risk Analysis

Once the risk is determined, it needs to be assessed according to the defined risk assessment method. Senior managers must devote themselves to risk assessment methods and apply them uniformly in different projects' risk assessments within the organization. So, how to accurately assess the risk? Individuals decide which scoring model to choose, qualitative and/or quantitative; they consider scoring the two parts of risk: The probability (probability) that a particular risk may be realized, and If it has an impact on the target (potential impact). SRA is an essential aspect of SRM. At this stage, risks are identified and then classified. After the risk is classified, the level of possibility and impact of the risk be analyzed. After checking the possibility of risks caused by various technical conditions, the probability is defined as a percentage. This methodological situation can be:

- ✓ The complexity of technology
- ✓ Practicalawarenessovercome by the testing team
- ✓ miscommunication within the team
- ✓ The team is distributed in a larger geographical area
- ✓ Use an old testing tool

For impact, the consequences of risk once occur. Understanding the impact is very important because to understand how the company is affected:

- ✓ What is the loss to the customer?
- ✓ What kind of loss the enterprise suffers

- ✓ loss of reputation or harm to society
- ✓ Money loss against legal actions against the company canceling the business license

SRA takes the assessment further. When analyzing risks, want to look at the identified risks and determine the damage they might cause. To analyze risks, you not only need to consider the potential events that may occur, but also the harm caused to them using qualitative or quantitative analysis. To analyze risk, you need to combine the likelihood of an event with its impact. SRA is the procedure of determining the possibility of risks in a software project. It investigates uncertainty and its impact in terms of timetable, quality, and cost (if it does occur). The two methods of analyzing risk are quantitative and qualitative. However, it is essential to know that SRA is not a precise science, but more like an art. Risk assessment usually takes a quantitative assessment, which aims to identify risks and quantify them on a numerical scale (such as 0.0 to 1.0 or 1 to 10); and qualitative, which depends on overall risk Impressions to identify them. Qualitative risk analysis uses terms, such as low to high, and medium to high rather than numerical values.

3.1.3.1 Qualitative Software Risk Analysis

Qualitative software risk analysis refers to the action determining the priority of risks for detailed analysis or action. For this, first, determine the possibility or likelihood of each risk and evaluate its effect on the success of the project. The commonly used scale is ranked from 0 - 1. Also, if the possibility of a risk occurring in the project is 0.5, then there is a 50% probability of occurrence. Also for ranging 1-5, five of which have the greatest impact on the project. The risk is then classified as source-based or effect-based. QRA is advantageous since you can not only reduce the ambiguity in the project but can also focus on high-impact risks, for which you can plan appropriate mitigation measures. Table 910 shows the input of risk analysis, the input tools and techniques, and the techniques performed. Risk Priority can be determined by evaluating and combining the likelihood of risk occurrence and risk impact for further measure

Table 10. Qualitative Software Risk Analysis

Inputs	Tools and Techniques	Output
➢ **Risk management plan** ➢ **Scope baseline** ➢ **Risk register** ➢ **Enterprise environment** ➢ **factors**	➢ possibility and risk impact assessment ➢ Probability and impact matrix ➢ Risk categorization ➢ Expert judgment	➢ Project documents updates

A)Components of Risk

In software risk analysis risk three main components are discussed in fig.4. i.e. risk, probability, and impact.

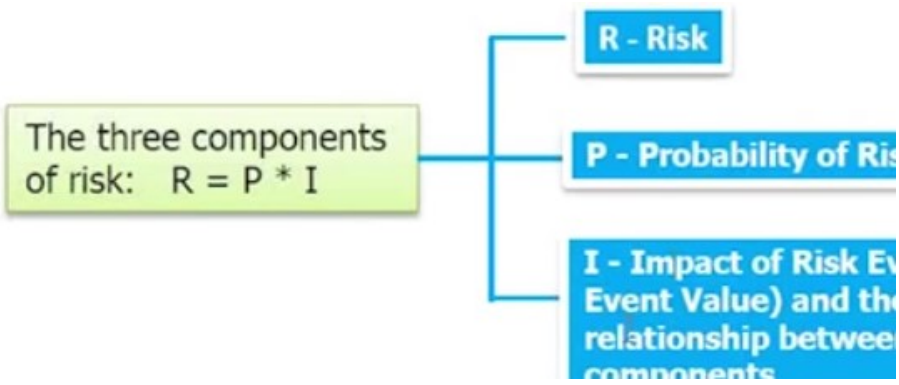


Fig.4. *Components of Risk*

The amount of risk is highest at- the start of the project and lowest at- the closing stages of the project

Fig.5. *Shows the Amount of Risk at the Beginning and the End*

a) Probability and Impact Matrix

This is the most common qualitative risk analysis tool. It can assess the probability (probability) that a specific risk occurs and the potential impact that a specific target may have. Analyze the possibility and risk impact of each risk and assign it to Nine point score:

- ✓A score between 1 and 9 or a Five-point scale: very low, low, normal, high, very high
- ✓A score between 1 and 5 and three points score l, m, h or 1-3 points.

Risk score = probability * impact

If the risk has a low probability and is assigned as 1, and the impact is significant and is assigned as 9

RS = 1 *7 = 7

In the probability and influence matrix, there are probability categories, and probability numbers are assigned to the probability, as shown in Table.11 and its description.

Table 11. Likelihood Scoring Guideline for Risk Assessment

Probability Category	Probability value	Description
Very high	9	Risk events expected to occur
High	7	Risk events are likely to happen
Probable	5	Risk events might or might not occur
Low	3	Risk events less likely than not to occur
Very low	1	Risk event not expected to occur

Table 12. Impact/Effect Scoring Guideline for Risk Assessment

Impact level	Impact value	Description
Very high	9	Very significant
High	7	Significant
probable	5	Significant and insignificant is equal
Low	3	insignificant
Very low	1	very insignificant

In table 13 briefly describes the sample impact scoring and describes the organization's process for evaluating, analyzing, prioritizing, and monitoring risks

Table 13. Impact/Effect Scoring With Project Risk Factors

Project Objective	Very Low 1	Low 3	Moderate 5	High 7	Very High 9
cost	Insignificant cost impact	<10% cost impact	10-20% cost impact	20-40% cost impact	>40% cost impact
Schedule	Insignificant schedule impact	<5% schedule impact	5-10% schedule impact	10-20% schedule impact	>20% schedule impact
Scope	Barely noticeable	Minor areas impacted	Minor areas impacted	Changes unacceptable to the client	The product becomes effectively useless
quality	Barely noticeable	Minor function impacted	The client must approve a quality reduction	Quality reduction unacceptable to the client	The product becomes effectively useless

This impact assessment focuses on the four core project management areas as cost, schedule, scope, and quality. The reason behind this is that cost, schedule, scope, and quality is the core function of project management [40]. The scoring mechanism is a popular software development organization and organizations used their mechanism to decide the level of the impact.

The relationship between chance and risk impact can be evaluated in fig.6.

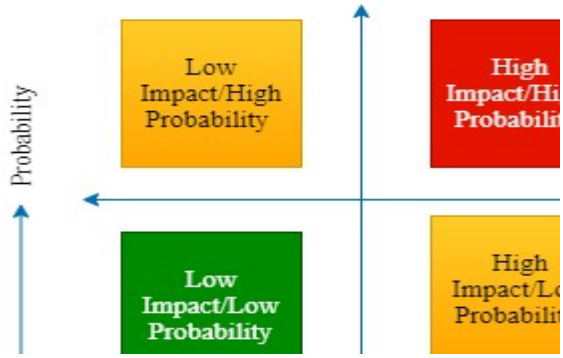


Fig.6. *show the probability and risk impact*

Table 14 shows that risk analysis uses very low, very high, low, high, and medium scores for probability and impact levels. The values are prioritized.

Table 14. qualitative risk level assessment

Probability	Very Low	Low	Medium	High
Very High	Medium	Medium	High	High
High	Low	Medium	Medium	High
Medium	Low	Medium	Medium	Mediu
Low	Low	Low	Medium	Mediu

3.1.3.2 Quantitative Software Risk Analysis

In contrast, quantitative Software risk analysis indicates the arithmetical analysis of the damage of known risks on the entire project. Quantitative risk calculates the possible outcomes of the project and calculates the possibility of still achieving the project goals. This helps decision-making, especially in the existence of ambiguity, and can set realistic cost, schedule, or scope targets. Quantitative methods for assigning values to risks require admittance to reliable statistical information to predict the likelihood of future risks. As stated earlier, a qualitative technique usually includes subjective measures as low, medium, and high[7]. These three measurements are the most popular in risk management. However, sometimes qualitative methods are more acceptable to management. In quantitative risk analysis, there are inputs, techniques, and tools shown in Table 415 for the inputs and outputs obtained by these tools and techniques

Table 15. Quantitative Software Risk Analysis

Inputs	Tools and Techniques	Output
➢ **Risk management plan** ➢ **Cost management plan** ➢ **Schedule management plan** ➢ **Risk register** ➢ **project environment factors**	➢ Data gathering and analysis techniques ➢ Expert judgment	➢ Project documents updates

Based on the scores of the likelihood and risk impact[21], the risk status can be determined as shown in Table 16 in color

Table 16. Quantitative Risk Level Assessment

Probability	1	3	5	7
9	9	27	45	63
7	7	21	35	49
5	5	15	25	35
3	3	6	15	21

When summarizing the two software risk analysis techniques clearly, it looks as shown in Table 17.

Table 17. Summary of QRA and Quantitative Risk Analysis

Qualitative Risk Analysis	Quantitative Risk Analysis
➢ **Performed first** ➢ **Should always be done** ➢ **Needed for a smaller project** ➢ **Quick and easy to perform** ➢ **The degree of all risk is undetermined** ➢ **Subjective**	➢ Performed after qualitative analysis ➢ Can be optional ➢ Needed for a large project ➢ Time-consuming ➢ The degree of all risk is determined ➢ Objective ➢ More detail

3.1.4 Software Risk Prioritization

Finally, the identified and analyzed risks are ranked according to their relative weight. Software projects usually contain a huge amount of risk factors. Therefore, it is not always possible to mitigate all factors. High-priority risks are considered first. Risk exposure and leverage be effectively mitigated to rank risks [27]

3.1.5 Software Risk Evaluation

Risk assessment includes risks that are highlighted based on defined risk type and their levels)[39]. RA is the procedure of recognizing and evaluating risks. SRE is a fundamental business practice that can be practical to industry, strategies, business agreements, plans, projects, and operations. This stage involves dividing the risk into high risk, medium risk, low risk, or tolerable risk. Besides, this stage is supported as the base for the next stage (the risk treatment stage). RA includes some technical analysis to:

- ✓ Measure hazards by applying administrative, technical, and environmental control measures.
- ✓ Transfer risks, such as buying insurance.
- ✓ Accept the risk, because it may be hard to mitigate the risk.
- ✓ avoiding risks by stopping business activities

Risk identification is an iterative process involving the project team, stakeholders, other managers involved or participating in the project, and finally individuals. Therefore, the risk should be determined and their occurrence should be estimated. Risk analysis includes identifying, analyzing, planning, monitoring, and resolving risks. In a software, project is expected that a certain degree of uncertainty will be generated in the estimates for reliable products. Many techniques for estimating the time and effort required for software development are completed in a limited time with limited resources. To analyze the impact of risks found in software development, project managers must identify risks. Software risk components are performance risk, support risk, cost risk, and schedule risk [34].

3.1.6 Software Risk Mitigation

In anSP, risk mitigation indicates a sub-stage of risk assessment [39] which is handled in a software project to avoid or transfer risks. RM deals with reducing the effect of risks by minimizing the chances of their occurrence. Risk mitigation can lead to risky items being eliminated or otherwise resolved [45]. Mitigating risks does not mean eliminating activities that generate risks. On the contrary, risk mitigation minimizes the effect of risk to a tolerable level [48]. Reducing risk involves choosing the option that best strikes a balance between performance and cost. In the entire system life cycle, different short-term and long-term mitigation methods may also need to be adopted [2]. Reducing the risk by reducing its impact to make it more acceptable to the project or organization can be called risk reduction.
Determine several mitigation strategies as a risk response plan. From the viewpoint of risk reduction, the communication between the team and the direct stakeholders in the organization is more relevant. Due to the different nature of the complexity and scope, a different process or stricter, specific tools and techniques require extensive use of well-known practices [37]. After categorizing and measuring the risks, the company can decide which risks to eliminate or minimize, and how many core risks to retain. Risk mitigation is achieved through a variety of processes, however, it should be based on the risk assessment completed in the previous step and prioritize the most important and dangerous risks.

3.1.7 Software Risk Documentation and Communication

As a PM only identifying, planning, and resolving risk events are not enough. A project folder or file needs to be created at the last part of each project to increase transparency and understand the project's timeline, workflow, and risks. This document summarizes the above reports and adds references to insights and best practices on how to handle risks. It supports other managers in understanding the input and output of projects similar to this. Communication between software project teams and stakeholders is the most important factor for successful risk management. While the communication arrangement in the organizational environment is weak problems, risks, and crises may arise[49]. Communication between appropriate organizational entities is necessary to correctly identify, analyze plan, track, and control risks [48]. Risk communication is the basic idea of the paradigm to emphasize its universality and importance. Maintaining a risk record enables customers to outlook evolution and to know detail about risk. However, if the team just has a written risk assessment, and hasn't gone through each of the steps, then it's not likely to comply with the law at all.

3.1.8 Review the Assessment and Revise it if Necessary

Few work environments stay the same, so review and modification of RA are important when needed. Environments are forever changing, so the risks are also changing. With the introduction of new equipment, new technology, and new personnel, each item brings new risks. Continuously check and update the RA stages to grasp these new hazards. Nothing can stay the same forever. Managers and teams must be responsible for monitoring processes and developing reporting procedures, discussing and helping implement solutions, and monitoring the effectiveness of solutions. The RA must be reviewed frequently to check that risk is correctly assessed. If any changes that may increase the effect of the software product occur through the development cycle, such as changes in requirements, technological progress, development models, and time constraints, they should also be reviewed. Risk assessment review has no time limit.

The team is accountable for deciding when the review is needed, however, the RA is a working document, and as the development process changes, this information should be documented and modified. It is suggested that the risk assessment be reviewed at each stage of development. A risk mitigation plan can be regularly evaluated by the team to ensure that it is comprehensive and effective. Every step in the plan needs to be reviewed and approved. If necessary, make further additions or modifications. A practical risk management technique establishes the most effective barrier to threats. Therefore, any resources that use software resources should be regularly checked for risk. The typical timeline for repeated risk assessments is to review the strategy at least once every two years. Any other assessments to assess emerging risks are done as required. This risk assessment may need improvements due to assessment tools and techniques. Thus, it's probably time for an update! Remember that, one of the reasons risk assessments are important is to make improvements. Requirements change, people change, and new data and best practices develop over time. The appropriate risk assessment framework is developed in this (see fig 3.5).

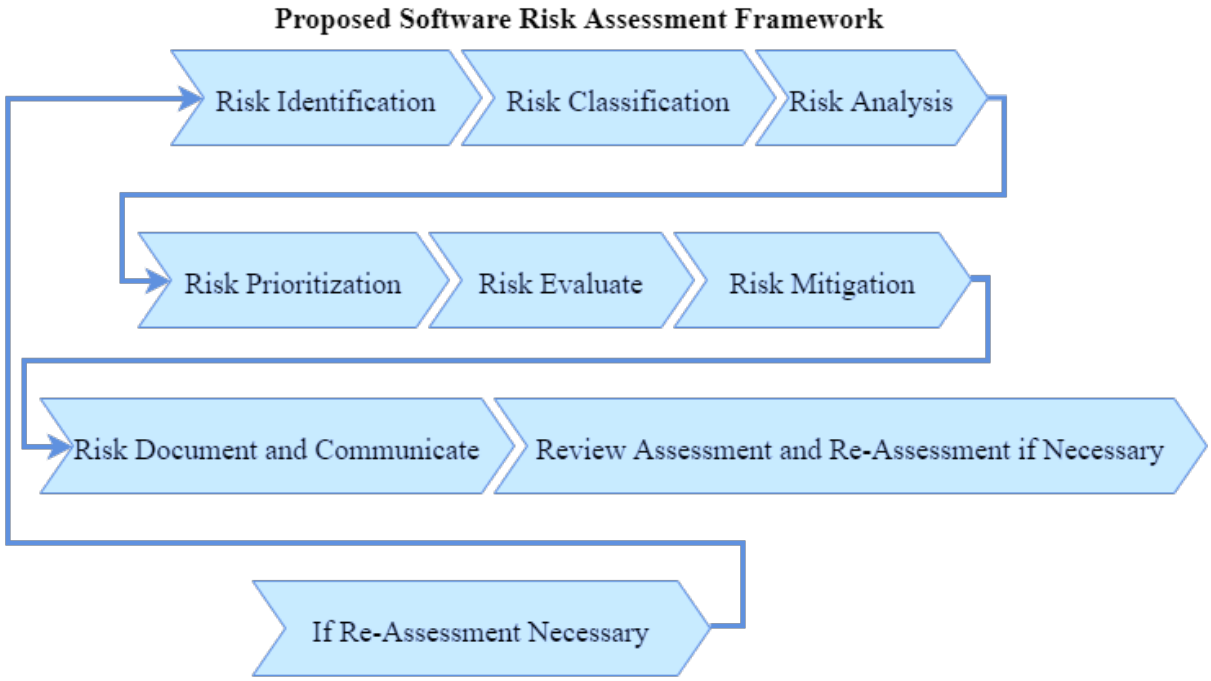


Fig.7. *Proposed SRA Framework*

3.2 Data Collection

The data is collected from the PROMISE repository software engineering data. The data set is COCOMO NASA93 public data provided by PROMISE project data points [12]. Here, classify attributes according to the risks that occur. Especially "size", 15 COCOMO-I multipliers, "workload" and 7 attributes describing other characteristics of NASA software systems (project ID, project name, application category, flight/ground system, NASA center, "year") Completion, and development mode). Please note that "workload" is measured in a calendar month of 152 hours, including development and management time. To demonstrate, start with a simple linear model and demonstrate how to identify risks and make improvements. A common mistake is to use ordinal values for the nominal ratio. For example, consider the "NASA Center" attribute (Sarcià, 2009).

Table 18. Dataset Attributes with a Brief Description

S No	Risk Occurred	ID	Attribute Name	Purpose
1	Personal Risk	ACAP	Analysis Capability	It describes the capability of analysis. Its rating should not consider the level of experience
2		PCAP	Programmer Capability	The evaluation depends upon the capability of the programmer as the team not individual
3		AEXP	Analysis Experience	Rating should be done on the level of application the project has experienced.
4		VEXP	Virtual Machine Experience	It describes how an effective programmer can use the hardware
5		LEXP	Language Experience	It measures the level of programming language and software tool experience of the project team.
6		MODP	Modern Programming Process	It describes the programming knowledge of the team members and rates them according to their level
7	Process Risk	Tool	Use Of Software Tools	Tool rating ranges from simple edit and code
8		TIME	Time Constraint	Percentage of accessible implementation time predicted to be applied by the software (system) to overwhelm the time of implementation.
9		TURN	Turnaround Time	This attribute defines how much time is taken to succeed in the project
10		DATA	Database Size	It captures the influence the test data requirements

				have on program development.
11		SCED	Schedule Constraint	It specifies a rating that represents the degree of schedule constraint imposed on the system software
12		STOR	Main Memory Constraint	It specifies a rating that represents the degree of main storage constraint imposed on the system software or subsystem.
13	Reuse Risk	VIRT	Machine Volatility	Describe the volatility of the machine, how it works effectively
14		CPLX	Process Complexity	It describes the rating of the complex process of system software
15		RELY	Required Software Reliability	The software must perform its intended function over some time
16	Project Risk	RECORD NUMBER	Unique Id	Software projects must have a unique identity for identification
17		PROJECT NAME	Project Name	Specify the name of the software project
18		CAT2	Category of Application	Indicates the category of software application developed
19		FORG	Flight or Ground System	Specify the system specification
20		CENTER	Which NASA Center	Describe the NASA center of the project
21		YEAR	Year of Development	Indicate the year the project developed
22		MODE	Mode of Development	Specify the mode of the developed software project
23	Project Risk	EQUIVPHYSKLOC	Equivalent Physical 1000 Lines of Source Code	Illustrates the source code of the venture in KLOC
24	Process Risk	ACT_EFFORT	Development Effort In Months	Indicate the actual effort needed to succeed in the project.

Here, describe the different risk attributes extracted from the NASA database. The values of this risk attribute range from "very low to very high". A risk attribute may belong to two different categories. As [31] pointed out that from the top of the intermediate COCOMO, 15 factors affect software projects. These factors are divided into 4 different categories:

Software Product
- ✓ Reliability Requirement
- ✓ Database Size
- ✓ Product Complexity

Computer System
- ✓ Execution Time Constraints
- ✓ Main Storage Constraints
- ✓ Virtual Machine Volatility
- ✓ ComputerTurnaroundTime

Hardware Resources
- ✓ Analyst Capability
- ✓ Virtual Machine Experience
- ✓ Programmer Capability
- ✓ Programming Language Experience
- ✓ Application Experience

Software Project
- ✓ Use of Modern Programming Practices
- ✓ Tools
- ✓ Required development Schedule

4. Experiment and Result Discussion

4.1. Assessment of Software Risk and Evaluation

As discussed in the previous chapters, software risk assessment can take place by two techniques; qualitative and quantitative analysis. However, this focuses on qualitative measurements of project risk factors which are written in linguistic terms. Those terms are appropriate to analyze the likelihood and risk impact that is caused by the probability. Furthermore, in software risk assessment, those linguistic variables are used in this study to assess software risk factors which may be categorized under software schedule risk, software platform risk, software product risk, software reuse risk, personnel risk, and software process risk. Table 19 briefly describes linguistic terms and their description.

Table 19. 1 Linguistic Term for risk impact

Linguistic variable	Description
VL	Very Low
L	Low
M	Medium
H	High
VH	Very High
XH	Extra-High

To assess the software risk from the proposed model 93 software projects data sets are used. The software projects data are shown in Table 20 where the qualitative value of considered software risk factors are represented in terms of low(L), medium(M), high(H), extra high(XH), very high(VH), and very low(VL). Table 20 shows the software project risk to identify the high risky or low-risk projects. The probabilistic assessment of software risk helps the software project manager in decision making for instance if the probability is: Low – very unlikely that the project leads to failure. Medium – There is a 50%–50% chance that the project leads to failure. High – The chances are very high that the project leads to fail.

Table 20. Software Risk Assessment Dataset

recordnumber	rely	data	cplx	time	stor	virt	turn	acap	aexp	pcap	vexp	lexp
1	h	l	h	m	m	l	l	m	m	m	m	h
2	h	l	h	m	m	l	l	m	m	m	m	h
3	h	l	h	m	m	l	l	m	m	m	m	h
4	h	l	h	m	m	l	l	m	m	m	m	h
5	h	l	h	m	m	l	l	m	m	m	m	h
6	h	l	h	m	m	l	l	m	m	m	m	h
7	h	l	h	m	m	l	l	m	m	m	m	h
8	h	l	h	m	m	l	l	m	m	m	m	h
9	h	l	h	xh	xh	l	h	h	h	h	m	h
10	m	l	h	m	m	l	l	h	vh	vh	m	h
11	m	l	h	m	m	l	l	h	vh	h	m	h
12	m	l	h	m	m	l	l	h	vh	vh	m	h
13	m	l	h	m	m	l	l	h	vh	m	m	l
14	m	l	h	m	m	h	l	h	h	h	l	vl
15	m	l	h	m	m	l	l	h	vh	h	m	h
16	m	l	h	m	m	l	l	h	m	m	m	vl
17	m	l	h	m	xh	l	l	h	vh	vh	m	h
18	m	l	h	m	m	l	l	h	h	h	m	h
19	m	l	h	m	m	l	l	h	vh	h	m	h
20	m	l	h	m	xh	l	l	h	h	m	m	h
21	h	l	h	m	m	l	l	m	m	m	m	h

The summary of the risk assessment is presented in Table 21 A total of 2232 risk factors are explored,15 risks are very low, 241 are low, 526 are medium, 465 have a high,121 have very high, and 26 are extra high. The majority of these risk factors are associated with medium and high software risk factors projects. This assessment can be done by using the **value_count** () function of the panda's library from the anaconda framework. This **value_count()** function counts the number of risk levels i.e. low, medium, high, etc.

Table 21. Summary of Software Risk Assessment

Risk Category	Risk Factors	Very Low	Low	medium	High	Very High	Extra-High
Personal Risk	acap	0	0	32	51	10	0
	pcap	0	0	43	40	10	0
	aexp	0	1	35	45	12	0
	vexp	4	14	52	23	0	0
	lexp	4	7	12	70	0	0
	modp	1	19	30	37	6	0
Process Risk	Tool	3	18	51	10	10	0
	time	0	0	54	13	18	8
	turn	0	42	29	20	2	0
	data	1	36	31	18	7	0
	Sced	0	34	40	19	0	0
	stor	1	0	49	10	20	13
Reuse	Virt	0	65	19	9	0	0

Risk	cplx	0	3	10	58	17	5
	rely	1	2	39	42	9	0
Total		15	241	526	465	121	26

In table 22 several software risk factorsare assessed and can determine which software risk factor can affect the software project or whether it is simple to make a decision depending on the result. The result shows that software risk factors like lexp, cplx, and virt affect the software project due to their risks. Figure 4.1-4.3 illustrates the level of risk of programmer Capability(pcap), Analysis Capability (acap) software risk factor, ,and machine Volatility (virt) software risk factor. Generally as presented in table 23 and drawn in the figures project manager can easily decide the project risk and can easily identify which risk factors can affect the project objective and easily make a decision. These figures are also plotted using **matplotlib.pyplot** library from the anaconda framework.

Fig.8. *Programmer Capability (pcap) Software Risk Factor*

Programmer capability (pcap) shows low risk on the project from the analyzed result. Thus it is easily measured by the development team.

Fig.9.*Analysis Capability (acap) Software Risk Factor*

Analysis capability (acap) shows high risk on a software project as analyzed from the result. Thus it must resolve first because it may damage the whole system.

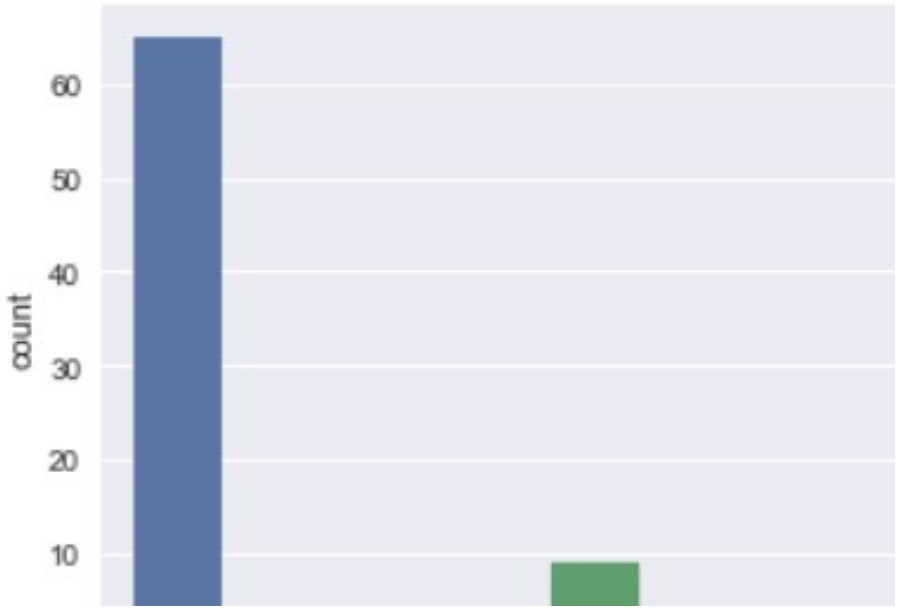


Fig.10. *Machine Volatility (virt) Software Risk Factor*

As stated in figure 4.3several factors have a high impact on the software projects especially factor ACAP have a high risk on the project as analyzed in fig whereas PCAP software risk factor has a medium risk and VIRT software risk factor has a low risk on the software project.

4.1.1 Importance of Software Risk Factors

Table 24 illustrates all risk factors which indicated the risk of the software project and determined which were the highest risk factors and very important. The software risk factors stor, vexp, data, time, rely, aexp, turn, virt, modp, and cplx were identified as very important factors whereas, software risk factors pcap, acap, and lexp factors were identified as important depending on their mean score value. This means the score value can be computed using the **.describe ()** function from pandas. The result can be used in the ranking of the importance of the listed software risk factors i.e. in order of importance is ranked as follow:stor,vexp,data, time,rely,aexp,turn,virt,modp,cplx,pcap,acap,and lexp

Table 22. Statistical Analysis of Risk Dataset

	rely	data	cplx	time	stor	virt	turn	acap	aexp	pcap	
count	93.000000	93.000000	93.000000	93.000000	93.000000	93.000000	93.000000	93.000000	93.000000	93.000000	93.
mean	1.193548	1.311828	1.010753	1.225806	1.548387	1.107527	1.139785	0.559140	1.150538	0.677419	1.
std	1.153992	0.884387	1.402596	0.795764	1.156420	0.540981	0.774476	0.683062	1.169785	0.662034	0.
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
25%	0.000000	1.000000	0.000000	1.000000	1.000000	1.000000	1.000000	0.000000	0.000000	0.000000	1.
50%	2.000000	1.000000	0.000000	1.000000	1.000000	1.000000	1.000000	0.000000	2.000000	1.000000	2.
75%	2.000000	2.000000	2.000000	2.000000	2.000000	1.000000	2.000000	1.000000	2.000000	1.000000	2.

Table 23 demonstrates the statistical measure that calculates the relationship or correlation between two factors. The values are in the range of -1.0 and 1.0. The computed result is >1 and < 1 there is a mistake in correlation in the analysis. This result can also be calculated by **.corr ()** from a pandas data frame. If the risk has a correlation coefficient of 1.00, their level would be moved in the same direction 100%. However, if the software risk factors have coefficients of -1.00 (-ve 1.00), their level of risk values are moved in the opposite direction, which means they move together 0% of the time.

Table 23. Correlation Coefficient of Software Risk Factors

	rely	data	cplx	time	stor	virt	turn	acap	aexp	pcap	vexp	
rely	1.000000	-0.176934	0.079286	0.366170	0.253548	0.331935	0.115342	-0.207733	-0.190910	-0.130801	-0.043045	-(
data	-0.176934	1.000000	-0.081597	-0.286478	-0.190277	-0.161720	-0.096070	0.373989	-0.129921	0.173670	-0.074745	(
cplx	0.079286	-0.081597	1.000000	0.192573	0.244276	-0.044516	0.238752	-0.210561	-0.020872	-0.007930	-0.335307	(
time	0.366170	-0.286478	0.192573	1.000000	0.726229	0.119730	-0.228141	0.125143	-0.095297	-0.107820	-0.095328	-(
stor	0.253548	-0.190277	0.244276	0.726229	1.000000	0.008968	0.046979	0.006658	-0.182215	-0.078774	-0.188803	-(
virt	0.331935	-0.161720	-0.044516	0.119730	0.008968	1.000000	0.326939	0.012019	-0.077385	-0.144894	0.022540	(
turn	0.115342	-0.096070	0.238752	-0.228141	0.046979	0.326939	1.000000	-0.231538	0.036509	-0.038296	-0.172197	(
acap	-0.207733	0.373989	-0.210561	0.125143	0.006658	0.012019	-0.231538	1.000000	0.165581	0.451267	0.267590	(
aexp	-0.190910	-0.129921	-0.020872	-0.095297	-0.182215	-0.077385	0.036509	0.165581	1.000000	0.554626	0.320732	(
pcap	-0.130801	0.173670	-0.007930	-0.107820	-0.078774	-0.144894	-0.038296	0.451267	0.554626	1.000000	0.269303	(
vexp	-0.043045	-0.074745	-0.335307	-0.095328	-0.188803	0.022540	-0.172197	0.267590	0.320732	0.269303	1.000000	(
lexp	-0.089160	0.148056	0.128138	-0.119775	-0.016898	0.054305	0.143493	0.252608	0.111252	0.184330	0.297235	1

Table 24. Calculated probability of the value of software risk

Machine learning algorithm	Risk probability		
	Low	Medium	High
SVM	0.21	0.23	0.56
	0.21	0.22	0.57
	0.03	0.23	0.74
	0.20	0.22	0.58
	0.21	0.22	0.57
	0.21	0.22	0.57
	0.21	0.23	0.56
	0.20	0.22	0.58
	0.21	0.23	0.56
RF	0.05	0.30	0.65
	0.07	0.22	0.71
	0.05	0.22	0.73
	0.08	0.30	0.62
	0.05	0.22	0.73
	0.04	0.39	0.61
	0.06	0.22	0.72
	0.05	0.45	0.5
	0.04	0.22	0.74
	0.11	0.30	0.59
	0.08	0.32	0.60
	0.07	0.37	0.56
LR	0.16	0.34	0.5
	0.19	0.31	0.5
	0.12	0.32	0.56
	0.15	0.33	0.52
	0.18	0.24	0.58
	0.15	0.35	0.5
	0.09	0.31	0.60
	0.10	0.40	0.5
	0.08	0.27	0.65
	0.13	0.27	0.6
	0.06	0.35	0.59
	0.12	0.32	0.56
	0.08	0.30	0.62
	0.12	0.38	0.5

5. Conclusion and Recommendation

5.1 Conclusion

The purpose of this study is to develop an enhanced risk assessment framework that significantly improves the assessment steps. The proposed SRA framework includes phases such as risk identification, risk categorization, risk analysis, risk prioritize, risk evaluation, recording risk, and communicating and reviewing the assessment and modifying if necessary. Risk identification identifies the source of the risk across the SDLC then the type of risk can be categorized. This categorized risk can be analyzed by two techniques qualitatively and quantitatively risk analysis. The analyzed risk can be prioritized and recorded accordingly. Finally, review the assessment at the closing phase. Maybe there is a change in process and re-assessment if necessary as much as possible. This study specifically examines the effect of software risk assessment on project success. Proper risk assessment plays a significant role in the success of software projects by increasing organizational income, logical solutions, system engineering, cost estimation, schedule development, and performance measurement. This paper raised four research questions and perform research activity.

The first research question was determining the factors which are basic for categorizing software project risk. These risk factors are clearly defined in the, especially technical risk factors are the main factors in determining the software project risk and they play a critical role in the success of software projects. The second research question was how to assess this software project risk properly. As discussed in the paper proper software risk assessment starts by identifying the source of risk at each phase of the SDLC, determining the type of major risks, analyzing the categorized risk in quantitative and qualitative

methods, prioritizing risk level, recording risk and finally review the assessment and update assessment if it is necessary.

5.2 Recommendation

As discussed in this study the main focus is a technical risk that hinders the software project objective. An interested future researcher could look for other software risks such as security risk factors that mainly damage the goal of a software project. In this paper, an enhanced software risk assessment framework was proposed. The organization could use the proposed framework to make practical decisions about the organization's future, i.e. Evaluating whether requirements are changing in the organization, assessing changes, knowing when new services are needed, achieving the organization's needs, measuring development tools, evaluating expert professionals in productivity and capability. However, there is no popular guideline for risk assessment that guides development teams. Thus the future investigation of this area is conducted on developing a proper risk assessment guideline for complete software projects.

Compliance with Ethical Standards

Conflicts of interest: Authors declared that they have no conflict of interest.

Human participants: The conducted research follows ethical standards and the authors ensured that they have not conducted any studies with human participants or animals.

Reference

[1] Albaqi, H., & Sadiq, A. A. (2018). INTEGRATED SOFTWARE PROJECT RISKS METHOD BASED ON PDF-ANN TECHNIQUES. (December).

[2] Avdoshin, S. M., & Pesotskaya, E. Y. (2011). Software risk management.

[3] Barghi, B., & Shadrokh, S. (2020). Heliyon Qualitative and quantitative project risk assessment using a hybrid PMBOK model developed under uncertainty conditions. Heliyon, 6(November 2019), e03097.

[4] Barghi, B., & Shadrokh sikari, S. (2020). Qualitative and quantitative project risk assessment using a hybrid PMBOK model developed under uncertainty conditions. Heliyon, 6(1), e03097.

[5] Bhujang, R. K. (2017). Recent trends of Risk Management in Software Development : An Analysis. 1(February), 33–43.

[6] Bilal, M., Gani, A., & Haseeb, M. (2020). RISK ASSESSMENT ACROSS LIFE CYCLE PHASES FOR SMALL AND MEDIUM. (February).

[7] Bilal, M., Gani, A., Liaqat, M., Bashir, N., & Malik, N. (2020). Risk assessment across life cycle phases for small and medium software projects. Journal of Engineering Science and Technology, 15(1), 572–588.

[8] Boodhun, N., & Jayabalan, M. (2018). Risk Prediction in Life Insurance Industry using Supervised Learning Risk prediction in life insurance industry using supervised learning. Complex & Intelligent Systems, (April).

[9] Capretz, L. F. (2013). Software Project Risk Assessment and Effort Contingency Model Based on COCOMO Cost Factors.

[10] Capretz, L. F., Hydro, L., Nassif, A. B., Capretz, L. F., & Nassif, A. B. (2012). Fuzzy-ExCOM Software Project Risk Assessment Fuzzy-ExCOM Software Project Risk Assessment. 2(2).

[11] Choetkiertikul, M. (2018). Developing analytics models for software project management.

[12] Christiansen, T., Wuttidittachotti, P., Prakancharoen, S., & Vallipakorn, S. A. (2015). PREDICTION OF RISK FACTORS OF SOFTWARE DEVELOPMENT. 10(3), 1324–1331.

[13] Darwish, N. R. (2018). An Enhanced Risk Assessment Framework for Software Development in An Enhanced Risk Assessment Framework for Software Development in Multiple Projects Environment. (November).

[14] Doval, E. (2019). Risk management process in projects. 30(2), 97–113.

[15] Elzamly, A. (2011). Managing Software Project Risks with Proposed Regression Model Techniques and Effect Size Technique. (March 2014).

[16] Elzamly, A. (2014). A Comparison of Fuzzy and Stepwise Multiple Regression Analysis Techniques for Managing Software Project Risks : Implementation Phase A Comparison of Fuzzy and Stepwise Multiple Regression Analysis Techniques for Managing Software Project Risks : Implemen. (June).

[17] Elzamly, A. (2016). Quantitative and Intelligent Risk Models in Risk Management for Constructing Software Development Projects : A Review Quantitative and Intelligent Risk Models in Risk Management for Constructing Software Development Projects : A Review. (March).

[18] Elzamly, A., & Hussin, B. (2014). (Analysis Phase) with Proposed Fuzzy Regression Analysis Modelling. (August).

[19] Elzamly, A., & Hussin, B. (2015). Modeling and Evaluating Software Project Risks with Quantitative Analysis Techniques in Planning. (January).

[20] Elzamly, A., Hussin, B., & Salleh, N. (2016). Top Fifty Software Risk Factors and the Best Thirty Risk Management Techniques in Software Development Lifecycle for Successful Software Projects. 9(6), 11–32.

[21] Entrepreneurial (Chemical) Engineering. (1983).
[22] Ghaleb, T. A., Alsri, A. A., Shabaneh, L., & Niazi, M. (2014). A Survey of Project Risk Assessment and Estimation Models. (December). https://doi.org/10.13140/RG.2.1.3497.7764
[23] Goyal, M. V. (2015). Optimization of Software Project Risk Assessment Using Neuro-Fuzzy Techniques Master of Technology Mukesh Vijay Goyal. (May).
[24] Hijazi, H., Alqrainy, S. K., Muaidi, H., & Khdour, T. J. (2014). Risk Factors in Software Development Phases RISK FACTORS IN SOFTWARE DEVELOPMENT. (September).
[25] Hu, Y., Yat-sen, S., Yat-sen, S., Liu, M., Xie, K., & Yat-sen, S. (2007). Software Project Risk Management Modeling with Neural Network and Support Vector Machine Approaches Jiaxing Huang Abstract Juhua Chen. (70572053), 1–5.
[26] Hu, Y., Zhang, X., Ngai, E. W. T., Cai, R., & Liu, M. (2013). Software project risk analysis using Bayesian networks with causality constraints. Decision Support Systems, 56, 439–449. https://doi.org/10.1016/j.dss.2012.11.001
[27] Islam, S. (2010). Software Development Risk Management Model- a goal-driven approach Shareeful Islam.
[28] Katende, N., Ann, K., & David, K. (2017). Implementing Risk Mitigation, Monitoring, and Management in It Projects.
[29] Kaur, K., Kaur, A., & Kaur, R. (2014). of Different Risk Management Model and Risk Knowledge acquisition with WEKA. 2(4), 26–35.
[30] Kecerdasan, I., & Ikep, P. (2008). A GUIDE TO PROJECT MANAGEMENT BODY OF KNOWLEDGE (4th ed.).
[31] Kinra, A. (2014). RESEARCH ARTICLE A Fuzzy Based Model for Software Quality Estimation Using Risk Parameter Assessment. 3(3), 807–814.
[32] Kumar, C., & Yadav, D. K. (2015a). A Probabilistic Software Risk Assessment and Estimation Model for Software Projects. Procedia Computer Science, 54, 353–361.
[33] Kumar, C., & Yadav, D. K. (2015b). A Probabilistic Software Risk Assessment and Estimation Model for Software Projects. Procedia - Procedia Computer Science, 54, 353–361.
[34] Kumar, S. A. M. S. (2015). Software Effort Estimation and Risk Analysis Using Artificial Intelligence Technique. 3(01), 55–59.
[35] Leo, M., Sharma, S., & Maddulety, K. (2019). Machine Learning in Banking Risk Management : A Literature Review. https://doi.org/10.3390/risks7010029
[36] Li, X., Jiang, Q., Hsu, M. K., & Chen, Q. (2019). Support or Risk ? Software Project Risk Assessment Model Based on Rough Set Theory and Backpropagation Neural Network.
[37] Lim, N., & Perrin, B. (2010). Risk Mitigation And Management Scheme Based On Risk Priority. Australasian Journal of Information Systems, 10(3).
[38] Manalif, E. (2013). Scholarship @ Western Fuzzy Expert-COCOMO Risk Assessment and Effort Contingency Model in Software Project Management.
[39] Pa, N. C., Khalefa, M. S., Ali, H., Alasad, A., Zmezm, H., & Technology, I. (2016). Journal of Soft Computing and Decision Support Systems A Proposed Risk Assessment Model for Decision Making in Software Management. 3(5), 31–43.
[40] PMBOK. (2000). PROJECT MANAGEMENT INSTITUTE A Guide to the Project Management Body of Knowledge. Management, 2004, 2000.
[41] Pradhan, S. (2015). Impact of Security Factors in Software Project Risk Assessment using Neural Networks Impact of Security Factors in Software Project Risk Assessment using Neural Networks. (May).
[42] Rahman, M. S. (2018). Risk Management and Measurement of Risk Management Performance in Complex Projects. (July).
[43] Roy, B., & Dasgupta, R. (2015). A on Software Risk Management Strategies and Mapping with SDLC A on Software Risk Management Strategies and Mapping with SDLC. (November). https://doi.org/10.1007/978-81-322-2653-6
[44] Software Intelligence for Digital Leaders. (2019). Risk Management in Software Development and Software Engineering Projects. Retrieved from https://www.castsoftware.com/research-labs/risk-management-in-software-development-and-software-engineering-projects
[45] Ssempebwa, R. K. (2015). PROJECT RISK MANAGEMENT BY Eng Ssempebwa Kibuuka Ronald. (March).
[46] Stephanie Ray. (2020). The Risk Management Process in Project Management - ProjectManager.com. Retrieved from https://www.projectmanager.com/blog/risk-management-process-steps
[47] Suresh, K., & Dillibabu, R. (2018). Designing a Machine Learning Based Software Risk Assessment Model Using Naïve Bayes Algorithm. TAGA Journal, 14, 3141–3147.
[48] Tamilnadu. (2018). Designing a Machine Learning Based Software Risk Assessment Model Using Naïve Bayes Algorithm. 14, 3141–3147.
[49] Wanderley, M., Menezes, J., Gusmão, C., & Lima, F. (2015). Proposal of risk management metrics for multiple project software development. Procedia - Procedia Computer Science, 64, 1001–1009. https://doi.org/10.1016/j.procs.2015.08.619
[50] Yet, B., Constantinou, A. C., Mary, Q., Fenton, N. E., Mary, Q., Neil, M., & Mary, Q. (2016).