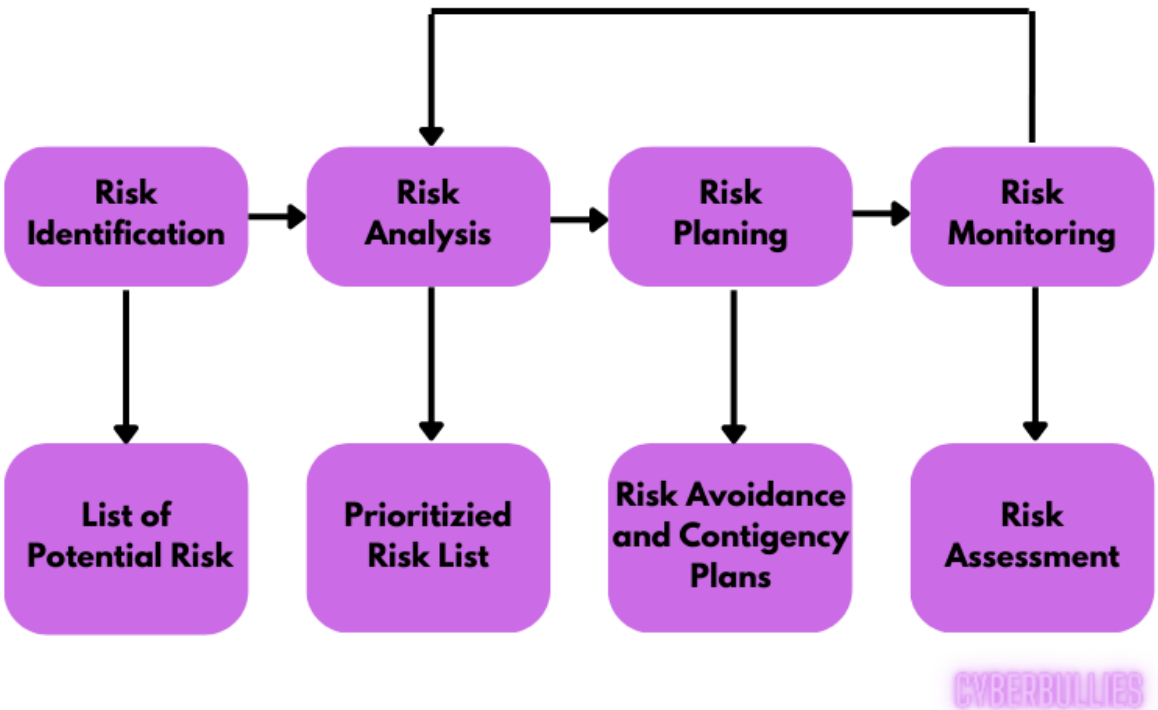
ICESHU4
Risk Management Report

1 Introduction

The Risk Management Report is divided into three components and one supporting document, the Risk List. Detailed descriptions of the potential dangers and the information in the Risk List document are provided in the second part. The third component is a record of the dangers we encountered while developing the product. Throughout the course of the project, both of these lists are updated.

A risk is the likelihood that some unfavorable event will take place, and in software engineering, it refers to a prospective or probable issue. In other words, risk is a concern that could have a negative impact on the performance or success of the system we are designing. The risk is the likelihood that the project or project-related stakeholders will suffer harm. Risk, however, is a necessary component of the software development process. Project risks have an impact on the resources or timeline; Product risks have an impact on the functionality or quality of the software being developed; Business risks have an impact on the company developing or acquiring the software.

Identifying and gathering potential risks in advance is essential. Because using this strategy, we might have calculated the likelihood that our project will experience unanticipated or undesired events. Therefore, it becomes necessary to effectively assess and manage any potential hazards. We come across risk management as a means of satisfying these demands. Limiting and minimizing identified risks is known as risk management. It focuses on locating hazards and creating strategies to lessen their impact on a project. Because risks have an impact on the organization generating or acquiring the software, the timeline, the resources, the quality or performance of the application that is being built, etc. To begin with, all potential hazards must be determined. The steps that follow should be taken in the proper order for risk management: **risk identification**, **risk analysis**, **risk planning**, and **risk monitoring**.



Risk Identification: In this step, potential risks that can affect the project or company are identified and recognized. Risks can be internal (such mistakes made by people or broken equipment) or external (like shifts in rules or the state of the economy). The objective is to compile a thorough list of probable risks.

Risk Analysis: Once a risk has been identified, it is evaluated for possible impact and chance of occurrence through risk analysis. This step entails evaluating the seriousness of each risk, comprehending its potential repercussions, and calculating the likelihood that it will occur. As a result, risks are prioritized and resources are allocated appropriately.

Risk Planning: Risk planning is creating strategies and action plans to reduce or address hazards that have been recognized. This step focuses on developing specific countermeasures to lessen hazards' possibility or effect. It could entail risk avoidance (complete risk elimination), risk transfer (risk transferring to a third party, such as insurance), risk reduction (implementing preventive measures), or risk acceptance (accepting the risk and making preparations for it).

Risk Monitoring: Throughout a project or business operation, identified risks are tracked and reviewed as part of an ongoing process called risk monitoring. It entails monitoring changes in the external environment, periodically reassessing risks, and assessing the efficacy of risk mitigation measures. This procedure permits prompt modifications or interventions when they are required, assisting in the proper management of hazards.

2 Description

This section covers the potential risks that might occur during the project's execution as well as the urgent measures that should be implemented in their event. The following risks are ranked according to their severity.

1- A project member may have a problem that she/he cannot attend the course.

Mitigation Strategy: When a member is unable to take the course, his/her contribution to the group should be determined and someone else should be found to fulfill these duties completely.

2- A team issue could arise for project members.

Mitigation Strategy: Requirements and responsibilities should be clearly defined to avoid in-group conflicts and arguments.

3- While the project is still in progress, a team member unexpectedly leaves the group.

Mitigation Strategy: Customers and developers should be informed of any potential problems and delays. The contributions and responsibilities of the leaving member should be determined and these responsibilities should be given to other members.

4- The development team members are struggling from illnesses.

Mitigation Strategy: To minimize the effects of the team member's absence, rearrange the tasks and the team.

5- The time needed to finish the software has not been given significant consideration.

Mitigation Strategy: Request that the project manager create a basic timetable. Inform the stakeholders of the situation as needed, put more work into developing the project, and ask for deadline extensions.

6- Indecision and disagreement about the work to be done in the details of the project.

Mitigation Strategy: The organization has been reorganized so that various management is in control of the project.

7- Customers may submit negative feedback on its design and design, and changes may be requested in requirements that require major design revisions.

Mitigation Strategy: Identify what kind of design the client needs. Ask questions to be sure you understand. Keep your dependencies to a minimum while working on the design. Obtain traceability data to maximize the information hidden in the design and evaluate the impact of requirements on change.

8- The group members fail to fully understand the customer's requirements, and the requirement descriptions are weak and ambiguous.

Mitigation Strategy: By speaking with the customer, the needs can be fixed and clarified without having a negative impact on any other requirements or the progress of the project. To resolve the danger, the group must discuss every necessity. It is assured that all project needs are thoroughly understood by all group members by conducting a thorough interview with the clients who establish the requirements and project specifications.

9- One or more of the project's requirements are changed at the customer's request after implementation.

Mitigation Strategy: The stakeholders should be informed of any potential delays and the effects of the modified requirement.

10- It's possible that the used software tools don't integrate properly.

Mitigation Strategy: A plan describing the software tool integrations should be created first, and then it is important to ensure that the software tools that can be included function as an integrated system in order to quickly remove any potential incompatibility.

11- Because the planned technology fails to satisfy the project's needs or cannot execute essential requirements, it is replaced with new technology, from which the system is needed.

Mitigation Strategy: Select a technology(framework, library, component) that will be current for a long time and is better suited to the project's specifications and content. As soon as you can, try to upgrade the old framework to the new framework. Use current software tools to transform the project if necessary.

12- There are no available courses for a group member who needs them.

Mitigation Strategy: Group members can learn more about topics about which they are less aware by receiving help and advice from group members who are more informed about such topics.

13- The size of the software project didn't get enough consideration.

Mitigation Strategy: Write efficient code, if at all possible, make use of compression methods, and then improve the system's storage.

14- The database may not be able to handle the expected number of transactions.

Mitigation Strategy: Look for approaches to less transactions. Check into the idea of replacing the current, underperforming database with one that performs better. Analyze thoroughly what occurs in change.

15- In the event of a disk failure caused by running out of space on the disk or by a production error, the information stored in the database is deleted or destroyed.

Mitigation Strategy: Database disks should be stored on different disks and backup or RAID systems should be used. The deleted or destroyed database should be restored using these backup techniques.

16- There is a failure to implement one essential use-case for the demo.

Mitigation Strategy: Customers should be informed of the error. If you are able, request for some extra time.

17- Customers are unable to fully understand the significant effects of the changes that will take effect as a result of requirements modifications.

Mitigation Strategy: Try to arrive at an understanding with the clients that the modifications shouldn't have a big impact on how the project is run. Create a document to inform clients of the changes based on changed needs. The project schedule is modified to take into account the recently added requirements.

18- Security vulnerabilities may exist in the project.

Mitigation Strategy: Necessary implementations should be made. A reliable architect and design must be established.

19- There are no specifications for required interfaces included in the schedule.

Mitigation Strategy: According to a predetermined timeline, all critical interface standards must be complete, consistent, and verifiable.

20- Overlapping constraints caused by various requirements complicates implementation.

Mitigation Strategy: It is possible to make architectural modifications to ensure all needs are met without ignoring any of them. It should be researched how other products with similar characteristics manage these interlocked constraints. The client should be called if the issue isn't fixable in the same way. In the established order of priority, the less essential requirements should be compromised.

21- The defect repair rate wasn't given serious consideration.

Mitigation Strategy: For each potential defect, have a specified enough repair time and resolution process that works into the program development timetable.

22- Too much time gets spent on development tasks.

Mitigation Strategy: Reduce the duration of time required to finish the task by using strategies like pair programming and extreme programming. If required, assign tasks to extra team members.

23- A group member's workforce is overstated. There is a poor fit between the amount of work given and the workforce of a group member.

Mitigation Strategy: Re-assign the tasks to the group members according to their assigned workforces. As a consequence, each group member is given tasks depending on their own workforce.

24- A member of the development team misunderstands a task that has been assigned to him.

Mitigation Strategy: Make sure team members are communicating effectively with each other and review other commits to mitigate this.

25- A development task has conflict with another development task to be completed.

Mitigation Strategy: Use programming techniques like pair programming or working in different branches to mitigate the impact of the situation and inform the team member in charge of the task.

3 Risk Management Report Specifications

The table below shows problems which we encounter during developing the project and our strategies to overcome these problems.

Problem	**Strategy**	**Status**
The time required for delivery is underestimated.	We rearranged our schedule to finish delivery.	Risk Mitigated
Adaptability of chosen tools.	Some tools that we will use for the project are hard to adapt. We changed some tools with more adaptive ones.	Risk Mitigated
Conflict between Members	We had disagreements about design. We discussed and chose the most liked one.	Risk Effect Minimized
Project Planning Issues	We tried to assign parts of the delivery in the most equal way.	Risk Mitigated

Some informations about Risk List document are as following:

Risk types:

P - People

R- Requirements

E - Estimation

Te - Technology To - Tools

O - Organizational

Impacts:

5- HIGHEST IMPACT - Catastrophic

4 - Serious

3- Moderate

2 - Tolerable

1- LOWEST IMPACT - Insignificant

Probabilities:

%25 - Unlikely

%50 - Moderately

%75- Highly Likely

Magnitude: Multiplication of impact and probability.

Owner:

Project Manager - Developer: Numan Kafadar

Software Architect - Developer: Mustafa Çağrı Korkmaz

Software Analyst - Developer: Osman Faruk Derdiyok

Software Tester - Developer: Umut Güngör

Product & Configuration Manager - Developer: Yunus Emre Terzi

Traceability Table

Traceability Table	Umut Güngör	Numan Kafadar	Mustafa Çağrı Korkmaz	Yunus Emre Terzi	Osman Faruk Derdiyok
Introduction					X
Description					X
Risk Management Report Specifications	X				
Information for Risk Documents	X				X