SOEN 390: Software Engineering Team Design Project
Winter 2021

Risk Management Plan

Submitted to Dr. Yann Gueheneuc

Team 9

Concordia University

February 3rd, 2021

TABLE OF CONTENTS

RISK CATEGORIES

The risks shall be divided in the following categories:

Risk categories	
Schedule Risk	Risks that are related to the schedule. These include and not limited to: - Incorrect time estimation - Improper resource allocation - Change in project scope - Unidentified complex functionalities and time to develop them - Volatile requirements
Budget Risk	Risks that are related to the budget. These include and not limited to: - Incorrect budget estimation - Improper tracking of budget - Wrong budget allocation - Cost overrun
Operational Risk	Risks that are related to the operations. These include and not limited to: - Insufficient resources - Lack of communication in the team - Wrong responsibility identification - No resource planning
Technical Risk	Risks that are related to the technical aspect. These include and not limited to: - Volatile requirements - Lack of technology availability for solution machine - Lack of skilled workers - Difficult project modules integration
Programmatic Risk	Risks that are related to the change of the plan. These include and not limited to: - Rapid market changes - Running out of funds - Change in government regulations - Loss of contract

Requirements Risk	**Risks that are related to the requirements. These include and not limited to:** - **Unidentified requirements** - **Late identified requirements** - **Inadequate requirements** - **Requirements that are not needed by stakeholders**
Design Risk	**Risks that are related to the design. These include and not limited to:** - **Improper architecture** - **Lack of OO features used in design**
Testing Risk	**Risks that are related to the testing phase. These include and not limited to:** - **Lack of representation of real-world scenarios with automated testing** - **False Positives** - **False Negatives**
Implementation Risk	**Risks that are related to the implementation phase. These include and not limited to:** - **Redesigned system** - **Change in technologies** - **New requirements added by stakeholders**

Risk Identification (TOP 10 RISKS)

Risk	Category	Description
Lack of time	Schedule	Not enough to complete tasks causing delays on the project completion
Lack of communication (team cohesion)	Operational	Team members do not report to each other and are not made aware of the progression of the project
Injection (hacking)	Technical	Hackers can attack the system by inserting malicious software such as SQL injections, DDoS attack, JS injections
Late identified requirements	Requirements	New requirements are introduced once the system is already in progress
Change in project scope	Schedule, Requirements	The project manager or the stakeholders change the requirements of the project, and therefore, the scope changes.
Improper architecture	Design	The choice of software architecture is not suitable to the system being developed.
Lack and improper testing	Testing	Testing is not performed frequently and does not cover a large part of the codebase.
Lack of reusable modules for software tasks	Implementation	The tech stack chosen does not offer many frameworks or libraries which requires developers to extra time building components.
Lack of object-oriented features	Design, Implementation	The software implementation of the system does not follow proper object-oriented features and design patterns which increase future maintainability effort.
Lack of resource planning	Operational	During the development of the project, the team realizes that they need a bigger budget, or more developers because they are not advancing in the project.

Risk analysis (Relate to impact matrix)

Risk	Impact	Probability	Risk exposure	Severity
Lack of time	0.5	0.25	0.125	Low
Lack of communication (team cohesion)	0.5	0.25	0.125	Low
Injection (hacking)	0.75	0.5	0.375	High
Late identified requirements	0.5	0.75	0.375	High
Change in project scope	0.25	0.75	0.1875	Medium
Improper architecture	0.75	0.25	0.1875	Medium
Lack and improper testing	0.75	0.75	0.5625	High
Lack of reusable modules for software tasks	0.25	0.5	0.125	Low
Lack of object-oriented features	0.25	0.5	0.125	Low
Lack of resource planning	0.5	0.25	0.125	Low

Risk Response (what will be done to avoid the risk)

Risk	Solution
Lack of time	Plan sprints before the beginning of the respective sprint. Also, start working on tasks as soon as the new sprint starts.
Lack of communication (team cohesion)	Mitigate by doing frequent meetings and making sure everyone is progressing
Injection (hacking)	Mitigate by preparing the website for SQL injections in the database
Late identified requirements	Accept, and reschedule the sprint to account for the new requirements
Change in project scope	Mitigate by taking time to elicitate the requirements and use the appropriate tools to interview the stakeholders, and making sure every possible information was extracted
Improper architecture	Avoid by consulting the software architect before starting and making sure that the proposed architecture is suitable for the ERP system.
Lack and improper testing	Avoid by doing unit tests for every method, and implement different code coverage techniques (mutation, data flow analysis, statement coverage)
Lack of reusable modules for software tasks	Avoid by selecting a tech stack which offers frameworks and libraries which can easily be reused, reducing development time.
Lack of object-oriented features	Avoid by using design patterns and using SOLID class design principles.
Lack of resource planning	Mitigate by doing frequent meetings to make sure the project has enough developers, designers, money and other useful resources