Project Risks

# Management

## Scope

### Gold plating inflates scope

The project team added their own product features that are not in requirements or change requests.

Mitigation: Discuss with the client/Jessie on the additional features the group would like to add.

### Estimates are inaccurate

Inaccurate estimates are a common project risk.

Mitigation: Discuss with Jessie, Prioritizing the user stories will help in identifying the tasks that should be finished beforehand.

## Change Management

### Inaccurate change priorities

When non-essential changes are prioritized impacting critical schedules.

Mitigation: Have everybody updated on the progress when you are working on the project. Inform the client of the sprint backlog for the following sprints at the end of each sprint.

### Change request conflicts with requirements

Change requests that make no sense in the context of the requirements.

Mitigation: Clarify with the client what is meant by the changes.

## Stakeholders

### Stakeholders become disengaged

When stakeholders ignore project communications.

Mitigation: Ask Jessie.

### Process inputs are low quality

Inputs from stakeholders that are low quality.

Mitigation: After the demonstration, if the client has not provided valid feedback, we could always ask specific questions on the features of the application.

## Communication

### Project team misunderstand requirements

When requirements are misinterpreted by the project team a gap develops between expectations, requirements and work packages.

Mitigation: We will hold client meetings with the client each sprint, but we should hold a stand-up meeting whenever we meet, and before we start with the project, we should clarify with one another what we understand from the requirements.

### Under communication

Communication is a challenge that is not to be underestimated. You may need to communicate the same idea many times in different ways before people remember it.

Mitigation: Whenever we hold a feedback session with Jessie, we can ask where our flaws lie, and what we could do to improve upon them.

### Impacted individuals are not kept informed

A stakeholder is missing in your communication plan. Anyone who is not informed but is impacted has an excellent reason to throw roadblocks.

Mitigation: Product owner has been tasked with updating the client on the progress of the group, and he should update the client after each sprint to notify the client of the sprint backlog.

## Resource & Team

### Learning curves lead to delays

When your project team need to acquire new skills for the project there is a risk that productivity will be low.

Mitigation: Whenever someone would require help, they should reach out to the group, and the group should always provide help.

### Training is not available

Quality training for certain skills can be difficult to secure.

Mitigation: The university provides us with free tools that we can apply to, such as LinkedIn Learning.

### Training is inadequate

Training is often a poor substitute for professional experience. Projects should not assume that resources will be fully productive in a new skill.

Mitigation: Discuss with Jessie whenever the group reaches a roadblock.

### Resources are unprecise

Resources who are negative towards the project may actively or passively sabotage project efforts.

Mitigation: Throughout the semester, we should follow the curriculum for the React side of the full-stack application, and we should clarify with the client whenever we intend to implement tools for the project.

### Resource performance issues

Resources who perform below expectations.

Mitigation: We will solve the conflict within the group, and when the conflict has not been resolved, we will discuss it with Jessie. Furthermore, the scrum master will have weekly check-ups on the team members.

### Low team motivation

Your team lacks motivation. This is a particularly common risk for long-running projects.

Mitigation: Since the motivation of the team seems to be lacking at the end of the semester, the team has agreed to put in the work at the start, so that at the end, the team would not have to put in as much work.

## Architecture

### Architecture lacks flexibility

The architecture is incapable of supporting change requests and needs to be reworked.

Mitigation: The group will make sure to ask the client in the first meeting how he would like the application to be.

## Design

### Design lacks flexibility

A poor design makes change requests difficult.

Mitigation: The group will follow the SOLID principles and will attempt to ensure clean code.

## Technical

*It is a good idea to skim over this section of risks.*

Mitigation: Read the OWASP top 10 and follow the information on how to prevent these risks.

### Technology components are not compliant with standards and best practices

Non-standard components that violate best practices.

Mitigation: When we hold the client meetings, we will ask the client for the practices he would like us to follow.

### Technology components have security vulnerabilities

Security vulnerabilities are key technological risks.

### Technology components are over-engineered

A component that is bloated with unneeded functionality and design features.

### Technology components lack stability

Components that crash.

### Technology components are not extensible

Components that are difficult to extend with new capabilities.

## Requirements

### Requirements are ambiguous

Requirements are unclear and open to interpretation.

Mitigation: Ask specific questions and discuss with the group.

### Requirements are incomplete

You can spot obvious holes in the requirements.

Mitigation: Clarify with the client what is meant with the requirements, and additionally, provide new requirements that fixes this problem.

## User Acceptance

### Users reject the prototype

One of the key methods of improving user acceptance is to get regular prototypes in front of users. There is always a risk that these prototypes will be rejected.

Mitigation: We hold demos for the client each sprint, and we will receive feedback that we will incorporate in the following sprint.

### User interface does not allow users to complete tasks

The risk that the user interface does not allow users to complete end-to-end tasks.

Mitigation: The team will ensure to test the application before we submit it to the client.