



# Risk Register

## Assignment 1 - GitHub Inspector

Team Nibble

Ayesha Ali - 29587778

Ryley Angus - 22647597

Shreya Kathuriya - 29506751

Akshith Patil - 29717973

1st September 2019



## Introduction

Risks can arise at any time during software development and they can have a negative impact on the delivery of the final product. Hence, our team have identified potential risks, estimated the likelihood and impact of each risk, and devised a corresponding monitoring plan and possible mitigation strategies.

## Risk Register

Risk Description	Likelihood	Impact	Mitigation Strategies	Monitoring Plan
Unavailability of team members	Very Likely	5	1. Schedule regular online meetings 2. Decide on a convenient time for weekly meeting during inception	Communicate with team members frequently to ensure all team members are on track and are available to attend the next meeting.
A team-member becoming ill	Likely	4	1. Eat well, drink water and sleep most days 2. If a team member is unwell, it's the team member's responsibility to notify the team as soon as possible, so the team can plan beforehand.	Task load on that team member can be reduced by allocating some of their tasks to other team members
Ambiguity regarding requirements	Likely	4	1. Discuss with the client to clarify scope or design of requirement. 2. Periodically verify implementation with client to minimise wasted development time. 3. Testers should also verify the suitability of their implementation against the provided requirement. 4. Actively listen during client meetings and take notes.	Feedback from Product owner during product review. This feedback may indicate if team members have misinterpreted the requirements

**Risk Register****Team Nibble**

<b>Risk Description</b>	<b>Likelihood</b>	<b>Impact</b>	<b>Mitigation Strategies</b>	<b>Monitoring Plan</b>
Client unavailability	Likely	2	1. Contact the client through email, phone call or other forms of communication to clarify any issues or uncertainties. 2. Schedule a face-to-face meeting as soon as possible.	Keep in contact with the client frequently and confirm that they are available for any meetings that require their input or presence.
Client Substitution	Unlikely	3	1. Do our best to minimise changes to requirements 2. Request the client to have a single representative, if possible, for the entire duration of the project.	Contact the client and ensure same representative is attending the meeting with the team.
Differing opinions within the team regarding design choices	Likely	3	1. Resolve the issue quickly. 2. Vote within the team after presenting the arguments for each choice.	Record and check the number of design issues raised during meetings.
Disgruntled team members sabotaging the project	Very unlikely	3	1. Be polite and respectful to each other. Resolve any disputes between members, with scrum master being the mediator 2. Don't antagonise each other. 3. Listen to, discuss and collectively consider all reasonable suggestions made by team members.	Monitor commits made by team members in Git, and any other contributions, regularly inspect all submitted work.
Enrolment changes for team members	Unlikely	4	1. Stay on task, keep up to date, discuss concerns with team members.	Communication is vital, ask team members to keep the team updated about any unexpected events

**Risk Register****Team Nibble**

Risk Description	Likelihood	Impact	Mitigation Strategies	Monitoring Plan
Minimal differentiation between team members ( <u>all</u> are junior software engineers)	Likely	3	1. Clarify the required roles as early as possible. 2. Discuss within the team who is capable of performing each role. 3. Assign roles as required. 4. Set expectations clearly.	Document assigned team roles, ensure team members are comfortable with their roles, check for any unplanned changes in team roles.
Natural disasters	Very unlikely	5	1. Monitor Emergency services to stay updated with any events.	Stay updated and stay connected with Emergency services
Poor communication between team members	Likely	4	1. Detail our preferred communication methods. 2. Remain polite & friendly. 3. Team members should be honest in describing the progress of their work 4. Ask the team members to actively listen during team discussion and ask question for any clarifications.	Ensure each team member communicates regularly and is not sparse in detail.
Technical issues with unfamiliar systems (Asana, GitLab)	Likely	4	1. Refer to existing documentation. 2. Ask team members for assistance. 3. Don't merge work if you think it may cause issues 4. Ask demonstrators for assistance	Regularly check the software being used for the project, be on the lookout for problems, eliminate issues quickly.
Team member inexperience with time management, JavaScript, unit-testing	Likely	5	1. Revise course material from earlier units (ENG1003) 2. Clear planning of tasks via Asana can assist with keeping team members on track.	Develop a spike. Eg: Have the whole team code a simple program, and inspect the code to make sure it is of high quality.