

Risk Assessment and Mitigation

Group 18

Team B

Olivia Betts
Zac Bhumgara
Nursyarmila Ahmad Shukri
Cameron Duncan-Johal
Muaz Waqas
Oliver Northwood
Teddy Seddon

Every project comes with risks, and in this document we lay out how we chose to manage them.

Our risk management process had 4 stages. First, we identified everything that could possibly go wrong. This resulted in many useful suggestions, as well as ones such as “a cyberattack takes out every computer in the room leaving us the only ones left standing”, which would be dismissed in the next phase.

We also categorised these risks based on what they might impact:

- Team, affecting the people working on the project
- Technology, affecting the project and what we used to build it. This got broken down into the subgroups of software and hardware
- Requirements, regarding what, exactly, the customer wanted us to implement
- Estimation, affecting the time we have available
- Project, affecting the project as a whole creation

Next, in the risk analysis stage, we assessed the likelihood of these risks occurring and the severity of their consequences. As this is only a relatively small project, we use a three point scale to assess these - high, medium, and low.

From there, we could start risk planning. We looked at what actions we could put in place to to (ideally) minimise and avoid these risks, or, if that wasn't possible, how to mitigate the damage caused, as well as implementing contingency plans if it all went completely wrong.

Finally, the risk monitoring phase. This isn't a static phase, more so one that the team members bear in mind whilst developing. This part requires vigilance, checking if any of the identified risks are showing themselves and putting into place the minimising measure, as well as keeping abreast of new risks we hadn't identified earlier.

Each identified risk has an owner, whether that's the whole team, a group of people, or just one. They are responsible for re-assessing the likelihood and severity of these risks, as well as reporting them.

These identified risks are presented below in a register. We gave each an ID, to easily refer back to them.

ID	Type	Description	Likelihood	Severity	Mitigation	Owner
R_WORK_UNDONE	Team	A member of the team is unavailable (extended illness, etc) and their work goes undone	M	M	Ensure there's at least one other person working on each section/able to take over	The whole team
R_INCOMPATIBLE	Technology - software	Game doesn't run on the customer's machine	L	H	Use Java, a cross platform software Ensure the final product isn't computationally expensive	Implementation
R_CHANGE_REQUIREMENTS	Requirements	Customer changes their mind about requirements - adding or removing some	M	M	Keep in regular contact with the customer	Requirements
R_DATA_LOSS	Technology - hardware+ software	Electrical fault results in data loss	M	H	Keep backups on both physical drives and the cloud	The whole team
R_COMMS	Team	Team communication breaks down/is ineffectual	L	L	Try multiple communication methods inc. in person	The whole team
R_OVERRUN	Estimation	The team runs out of time to complete project	M	H	Prioritise and ensure the most important sections get done Weekly Gantt chart to keep track of progress	Team leader, Methodology & Planning
R_FUTURE_PROOF	Project	Lack of documentation makes project hard to understand in the future	M	M	Follow coding standards re commenting Ensure important points are down in writing	Implementation (+ each other group documenting their own sections)
R_LIBRARIES	Technology - software	Lack of library documentation makes it hard to use	M	H	Chose library with established docs No person working alone	Implementation
R_MAJOR_BUGS	Technology - software	Major bugs render code unusable	L	H	Establish frequent tests Use best coding practices	Implementation
R_MIN	Technology	Small bugs negatively	H	L	As above and also:	Implement

OR_BUGS	y - software	affect user enjoyment			Resolve bugs when found, to ensure they don't create larger problems down the line	ation
R_SCOPE_CREEP	Project	Project morphs into something entirely different than required	M	M	Thorough planning frequently referred back to Iterative working to monitor state of project	Requirements + Architecture (to define project) Implementation (to stick to it)