Risk Assessment and Mitigation

Team 26

devCharles

Ross Holmes
Sabid Hossain
Thomas Pauline
Joel Paxman
Andrey Samoilov
Louis Warren

Introduction

One of the purposes for creating this document was to increase our “bus factor”. The necessity of a document like this is clearly shown in our own group:

During the beginning of our project we decided to start work on the website and part way through development we planned a meeting to develop it further, however a critical team member was unavailable. Due to the fact that we had already implemented this document and discussed mitigations we had three team members capable of understanding and continuing development which still made the meeting productive.

We created a risk document within our first few team meetings, discussed some immediate risks, how we would mitigate this and wrote it down. This was regularly added to when we held meetings twice a week in which we discussed progress and identified new risks; however we soon realised we needed to create a better system as the document grew. We decided to create a tabular system as we felt this was more succinct and readable than any other form (e.g. textual).

The risk assessment and mitigation document is therefore presented in the risk register, which is formatted as three tables:

- **Project** - these are risks that may affect the project's schedule or the resources used by the project (including team members).
- **Product** - these are risks that would affect the product quality or its ability to be fully completed e.g. tools in the project having bugs outside of our control.
- **Business** - these are risks that would affect the team's ability to procure/develop the software e.g. obsolete technology.

Each risk within each table is given:

- **ID** - Each risk has a unique identifier to allow for simple referencing and identification.
- **Description** - This is a description of the risk.
- **Likelihood** - This is how likely the risk is to occur - i.e. low, moderate, high.
- **Severity** - Should the risk happen this is an estimate of how much of an effect it would have again given as either low, moderate or high (Variable shows unknown impact).
- **Mitigation** - These are the steps we are taking to mitigate the risk, or steps to be taken in case of the risk occurring.
- **Owner** - This column indicates who is responsible for mitigating the risk and reporting an issue to the group if there’s a problem. This prevents risks from repeating (All team members are made aware of risks which also means that in rare cases where the owner doesn’t see the risk first the issue can be still found quickly).

As we continued with our meetings twice a week (generally), more risks were identified and added to the risk register. We then discussed their likelihood and severities, and categorised them. We worked together to come up with and implement ways in order to mitigate or avoid each risk. We would then assign an owner to each risk.

Something significant we did was to assign at least two people to all tasks deemed very important and we also (for these tasks) assigned a shadow role (somebody who had no responsibility for the task but was made aware so that they could provide aid for any issues). This meant our ‘bus factor’ was never below 2 and with important tasks never below 3. This meant we could be prepared for as many issues as possible and meant very few unexpected issues occurred.

Project Table

ID	Description	Likelihood	Severity	Mitigation	Owner
R1	A team member becomes unavailable due to illness or other issue	moderate	moderate	Make sure work is evenly distributed between team members and that everything has at least two people who are familiar with with each part of the project	Ross, Andrey, Louis
R2	The game engine becomes unavailable	low	high	Be aware of other game engines that are available to switch to	Louis
R3	Team member's knowledge of the codebase is not enough to contribute	high	moderate	Arrange a meeting to discuss the codebase with the team member	Andrey
R4	A file is accidentally deleted or corrupted	low	high	Keep updating the remote repository with github	Ross
R5	Something isn’t completed by the deadline set for it	moderate	moderate	Have regular meetings where the progress of the project is discussed	Louis
R6	Discord, our main messaging tool, goes down	low	high	Make sure we have multiple ways of contacting each other (e.g. via e-mails)	Joel
R7	A team member does not have the correct version of our dependencies, and therefore cannot contribute to the implementation	moderate	low	Make use of centralised dependency management (Gradle) and make sure everyone's base JDK is the same version	Andrey
R8	A team member’s computer breaks	moderate	moderate	Make sure everyone has access to another computer they can work on if necessary (e.g uni computers)	Tom
R9	The assets we used become unavailable	low	moderate	Be aware of places to get new assets from	Sabid
R10	Github goes down for a brief period (during a heavy development period)	low	high	Properly space out working times such that any impact over a certain period of time is minimised.	Joel
R11	A group member commits directly to main, losing changes made by others or introducing merge conflicts	Low	Variable	Lock the main branch to only accept github pull requests from other branches.	Andrey
R12	Fail to clarify everything during a client meeting	low	moderate	Contact the client to further clarify details.Prepare questions in advance in order to be sure of having every	Sabid

ID	Description	Likelihood	Severity	Mitigation	Owner
				required detail.	
R13	Team member's computer crashes losing unsaved changes	low	moderate	Regularly commit changes to the local source control system	Sabid
R14	Prerequisite code is unfinished, so sections cannot be completed	moderate	moderate	Prompt the owner of the required code to finish and begin a different section instead.	Andrey
R15	Requirements being added during Project progress (Scope Creep)	low	moderate	Get a clear understanding of what the client wants at the beginning of the Project	Everyone

Product Table

ID	**Description**	**Likelihood**	**Severity**	**Mitigation**	**Owner**
R16	The implementation has a progress halting issue	low	moderate	Help the team member resolve the problem	Andrey
R17	The game runs slowly on certain computers	low	moderate	Test the game on multiple computers with different specifications, and make changes if necessary	Joel
R18	The libraries used in the implementation don’t have/don't have enough documentation, and the team is struggling to implement them	high	moderate	Check documentation of libraries prior to picking them to be used in the project, if that's not possible look for alternative material (e.g. open source projects using the library)	Ross
R19	The game doesn’t resize well to some window sizes	moderate	moderate	Test the game in various common window sizes and make changes if necessary	Andrey
R20	Part of our code accidentally infringes upon someone's copyright	low	Variable	Depending on how the infringing code is licensed, the code can be kept as is with a copyright notice, or removed entirely if the licence is not compatible with the project	Tom

R21	Update to a library we are currently using that is incompatible with current code implementation	low	moderate	Only use libraries when required, and use libraries that are well known and available (trustworthy)	Ross
R22	Architecture does not support a required feature	moderate	high	Consider all possible architectures with their advantages and drawbacks.	Andrey, Louis, Ross
R23	Hosting service becomes unavailable	moderate	moderate	Prepare alternative hosting methods, including private servers.	Tom
R24	Compromising on design to complete function as quick as possible	low	moderate	Regularly check non-functional requirements during development.	Joel
R25	Lack of support for users post completion	high	low	Design simple to understand interface and provide clear instructions	Tom
R26	Bugs in code that are difficult to detect but appear with frequent use	moderate	moderate	QA / test to decrease frequency of issues	Louis, Ross

Business Table

ID	Description	Likelihood	Severity	Mitigation	Owner
R27	The software does not perform well on client's computer	moderate	high	Make sure we use optimised libraries and the codebase is overall efficient with its resources. Clarify with client about specifications of machines expected to run the game	Everyone
R28	University pauses teaching due to industrial action	moderate	high	Access alternative teaching material	Everyone