

Risk Assessment and Mitigation

Group 28

Piazza Panic

By OuseWorks (from devCharles, Group 26)

Ben Harris
Joshua Gill
Niamh Hanratty
Amy Raymond
Matthew Czyzewski
Matt Rohatynskyj

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:

There were times when not every member was available (e.g. due to illness) so ensuring risks were mitigated and people were assigned responsibility for such risks was necessary throughout the project to ensure progress could still be made.

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.. 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.

Updates to Risk Tables

We added a new column to show the dynamic aspect of the risks, i.e. when risks are more valid, or stop becoming risks, which link to our [weekly sprints](#).

Project Table

ID	Description	Likelihood	Severity	Mitigation	Owner	Relevant Dates
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	Matt Rohatynskyj and Joshua Gill	Increased likelihood between 17/03/23 - 17/04/23
R2	The game engine becomes unavailable	low	high	Be aware of other game engines that are available to switch to	All	N/A*
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	Matt Rohatynskyj and Joshua Gill	N/A
R4	A file is accidentally deleted or corrupted	low	high	Keep updating the remote repository with github	All	N/A
R5	Something isn't completed by the deadline set for it	moderate	moderate	Have regular meetings where the progress of the project is discussed	Amy Raymond and Niamh Hanratty	Increased severity nearing 03/05/23
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)	Amy Raymond	Increased severity between 17/03/23 - 17/04/23
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	Matt Rohatynskyj and Joshua Gill	N/A
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	All	Increased severity between 17/03/23 - 17/04/23

				computers)		
R9	The assets we used become unavailable	low	moderate	Be aware of places to get new assets from	Amy Raymond	03/03/23 - 03/05/23
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.	Ben Harris	03/03/23 - 03/05/23
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.	Ben Harris and Matthew Czyzewski	03/03/23 - 03/05/23
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 required detail.	Niamh Hanratty	N/A
R13	Team member's computer crashes losing unsaved changes	low	moderate	Regularly commit changes to the local source control system	Amy Raymond	N/A
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.	Ben Harris and Matthew Czyzewski	03/03/23 - 03/05/23
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	All	N/A
R16	Team members may not turn up to meetings.	moderate	high	Keep notes of our meetings so absent members know what we covered.	All	N/A
R17	A member of the team finds their role difficult and swaps with another team member	low	moderate	Check in with all team members to make sure they are happy with their assigned tasks	All	N/A

Product Table

ID	Description	Likelihood	Severity	Mitigation	Owner	Relevant Dates
R18	The implementation has a progress halting issue	low	moderate	Help the team member resolve the problem	Ben Harris and Matthew Czyzewski	03/03/23 - 03/05/23
R19	The game runs slowly on certain computers	low	moderate	Test the game on multiple computers with different specifications, and make changes if necessary	Ben Harris and Matthew Czyzewski	03/03/23 - 03/05/23
R20	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)	All	03/03/23 - 03/05/23
R21	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	Joshua Gill	03/03/23 - 03/05/23
R22	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	Ben Harris, Joshua Gill, Matthew Czyzewsk and Matt Rohatynskj	03/03/23 - 03/05/23
R23	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)	Ben Harris and Matthew Czyzewsk	03/03/23 - 03/05/23

R24	Architecture does not support a required feature	moderate	high	Consider all possible architectures with their advantages and drawbacks.	Amy Raymond, Joshua Gill and Niamh Hanratty	28/02/23 - 03/05/23
R25	Hosting service becomes unavailable	moderate	moderate	Prepare alternative hosting methods, including private servers.	Joshua Gill	03/03/23 - 03/05/23
R26	Compromising on design to complete function as quick as possible	low	moderate	Regularly check non-functional requirements during development.	All	03/03/23 - 03/05/23
R27	Lack of support for users post completion	high	low	Design simple to understand interface and provide clear instructions	Niamh Hanratty and Ben Harris	03/03/23 - 03/05/23
R28	Bugs in code that are difficult to detect but appear with frequent use	moderate	moderate	QA / test to decrease frequency of issues	Matt Rohatynskj	03/03/23 - 03/05/23

Business Table

ID	Description	Likelihood	Severity	Mitigation	Owner	Relevant Dates
R29	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	All	03/05/23
R30	University pauses teaching due to industrial action	moderate	high	Access alternative teaching material	All	N/A

***N/A indicates the associated risk was relevant throughout the entire project**

[1] - <https://www.northeastern.edu/graduate/blog/project-risk-management/>, 25/03/20, Scott W. O'Connor (TALKS ABOUT KEEPING RISKS UP TO DATE) accessed 21/04/2023