Risk Assessment and Mitigation

Team 14: Bass2

Katie Maison
Saud Kidwai
Jacob Poulton
Cody Spinks
Felix Rizzo French
Joachim Jones

5.(a)

Introduction and Process

This section will outline the risks that may occur in the project as well as how we can mitigate the effects of them. We will be using the risk management process outlined in the book *Software Engineering, Somerville Ian* [1]. Another process/outline that has been researched is the 'ProRisk Management framework' as described in *G. G. Roy, "A risk management framework for software engineering practice,*" [2] however this framework is above and beyond the scope necessary for the project.

Ian Sommerville describes the process of creating a risk assessment in 4 steps [1, p645].

- **Step one:** In this step we have discussed the risks and listed them in the table below. The process of this involved going through each stage/section of the project and identifying the possible things that could affect the project using the list of types or risks described by Sommerville as a starting point, as well as an additional section that wasn't listed which was those risks that come with the current Covid19 pandemic. [1, p647]

- **Step two** is to analyse the risk. As a part of this we have decided to describe the likelihood of risks as *'low'*, *'moderate'* or *'high'*. We feel this gives enough detail for the relatively small size of the project we are doing. Another option would be to add additional classifications such as 'low/moderate' or 'moderate/high' but this we feel is unnecessary and overcomplicates the classification. The consequences we will describe as *'Catastrophic'*, *'serious'*, *'tolerable'* or *'Insignificant'*. Alternatively we could classify them as 1-5 but using words is much easier to understand when referred to later.

- **Step three** is to create a plan for each risk to mitigate the effects of it if it were to happen. Where possible we will add multiple options for this to create an even more robust plan.

- **Step four** involves monitoring the risks and revising the assessment when more is known. To do this somebody will review this document every week, making relevant changes if necessary and updating the team on any changes that have been made and if a risk is becoming more likely.

The risk assessment will be displayed in tabular form in the following pages. This will take the format of 'ID', 'Risk category', 'Risk description', 'Probability', 'Consequence', 'mitigation' and 'notes'. The note section will contain the date that the risk was added and any assessments the person monitoring the risk has, for example 'the rates of covid are increasing. This risk probability has increased from 'moderate' to 'high''. This note section enables the team and customer to monitor the project in more detail as the risks evolve.

5.(b)   Risk Catalogue

ID	Risk Category	Risk Description	Probability	Consequence	Mitigation	Notes
000	Tools	Poor selection of platforms and engines	Medium	Tolerable	One way to mitigate this would be to test multiple platforms and engines at the beginning of the project, or alternatively to not commit to any platform decisions until they have been considered thoroughly.	
001	Estimation	Major bugs persisting into final versions of the product due to underestimation of time allocated for testing.	Low	Catastrophic	One way to mitigate this would be to implement test-driven development. This would ideally mean that major bugs would be avoided entirely, meaning only minor bugs would persist into potentially final products.	
002	People	Inconsistent coding practices throughout	High	Serious	One way to mitigate this would be to set up and agree on a coding standard that can be followed by everyone working on the project.	
003	Estimation	Technical debt* with underestimated avalanche causing the product as a whole to be more prone to breaking when being upgraded.	~~High~~ Medium	Tolerable	One way to mitigate this would be to keep a record of all the places where hack-solutions are used, and to not build upon areas where these are in place without first resolving them. This will increase the strength of the base code, meaning it is less likely to fall apart.	*Technical Debt is the concept of a build-up of weak code throughout a project that is set up with the "bit of a hack, but it works" mentality in order to save time in production. A little debt speeds development so long as it is paid back promptly with a rewrite [3] * Changed probability to medium as each member has now researched and understands the effects of technical debt prior to beginning implementation however not all people doing implementation have much Java/Libgdx experience (15/01/2022)
004	Estimation	A delay in one section of the project causing a knock on effect and a rushed or late submission	Medium	Serious	One way to mitigate this is the use of a review task in our planning. This means that while the rest of the team progresses, one or two people go back and finish anything that needs doing so the project is not delayed.	

005	Requirements	Misunderstand which features our customers truly care about	Medium	Serious	One way to mitigate this would be to ensure rigorous questioning and confirming were undertaken during interviews and other customer meetings.	
006	Technology	VCS failed and lost all work.	Low	Catastrophic	To mitigate this, GitHub will be used so that there is a local copy of the repository each time someone pulls the rep.	
007	COVID	A team member needing to isolate due to testing positive for COVID19	Medium	Tolerable	A way to mitigate this would be ensuring that there is a way of still participating in meetings remotely	There are many options for this, most likely we will use discord as the server is already set up
008	COVID	Local/National Lockdown resulting in inability to meet in person.	~~Medium~~ ~~High~~ Low	Serious	To mitigate this, all documentation will be stored on the cloud so it can be accessed remotely so that in the event that we cannot meet in person, there is no reduced access to material.	- 30/12/21 Updated to high risk due to new highest daily cases reported in the UK of 246,418 29/12/21 [4] - 16/01/21 updated to low risk due to statement by HSA that Omicron is relatively mild for most adults and this can be stated with good confidence [5]
009	Requirements	The addition/reduction of requirements midway through project causing severe delays due to difficulty editing implementation	Low	Serious	To mitigate this it is important that the codebase contains well documented and well laid out code.	
0010	Technology	Personal Internet Connection Failure	Medium	Insignificant/tolerable depending on how extensive	Have local backups and use Zoom for dial-in alternatives to join meetings.	
011	Tools	Significant update causing incompatibility	Medium	Serious/Tolerable	Keep previous versions of software for backtracking if the issues are serious.	
012	Technology	Loss of data/project due to corruption	Low	Serious	Use google drive to have a cloud copy of the data as it has version control builtin.	
013	Requirements	Misunderstanding requirements of project	Medium	Serious	Ensure we reread the briefing documents multiple times and attend all Q&A sessions	- Added 24/01/22: The rereading of the brief led to the realisation that a part of documentation had been missed

Environmental Risk Factors

There are a number of environmental factors related to the wider project as a whole. These can be broken down into: [6]

- Relationships between team members
- Constraints
- Team member Motivation
- Skill Set, Experience
- Organisation

Relationships between team members is the most important of these factors since a strong sense of team identity, commitment and trust is integral to every element of the project. Being strong in these areas will allow us to work fluidly and efficiently without having any internal issues that may waste time and instead allow us to have open discussions about how we feel certain parts of the project should be approached.

Furthermore there are **constraints** that may end up causing significant delays for the group. For example, clashing schedules over the Christmas holidays are likely to stop the full group meeting and slow down the development process of certain parts of the project.

In addition, constraints related to each team member, specifically their motivation, skill set and experience could affect not only how team members gel together but also how work is distributed amongst members of the team and how much weight each member is pulling.

Finally, **team organisation** is a crucial element of this project and if not done correctly could lead to a significant amount of lost time across the project's duration. For example, if there is not clear communication between who is doing what then we could have multiple people doing the exact same thing which would be a clear failure in planning. Instead we plan to use a number of software tools in order to optimise our communication and organisation and allow our work to be fairly and effectively distributed

Bibliography

[1] Sommerville, Ian. Software Engineering, EBook, Global Edition, Pearson Education, Limited, 2016. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/york-ebooks/detail.action?docID=5185655.

[2] G. G. Roy, "A risk management framework for software engineering practice," 2004 Australian Software Engineering Conference. Proceedings., 2004, pp. 60-67, doi: 10.1109/ASWEC.2004.1290458.

[3] W. Cunningham, "The WyCash portfolio management system," ACM SIGPLAN OOPS Messenger, vol. 4, no. 2, pp. 29–30, Apr. 1993, doi: 10.1145/157710.157715.

[4][UK Health Security Agency, "https://coronavirus.data.gov.uk/details/cases#card-cases_by_specimen_date," coronavirus.data.gov.uk. https://coronavirus.data.gov.uk/details/cases#card-cases_by_specimen_date (accessed Dec. 29, 2021).

[5] UK Health Security Agency, "SARS-CoV-2 variants of concern and variants under investigation in England," 2022. Accessed: Jan. 16, 2022. [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1048395/technical-briefing-34-14-january-2022.pdf.

[6] H. Khan, M. Mahrin, and S. Chuprat, "Environmental Factors Influencing Requirement Engineering in Global Software Development: A Review.," presented at the Proc. of the Second Intl. Conf. on Advances in Computer and Information Technology-, ACIT, 2013, [Online]. Available: https://www.researchgate.net/publication/265851676_Environmental_Factors_Influencing_Requirement_Engineering_in_Global_Software_Development_A_Review.