Risk Assessment and Mitigation

Original Assessment 1 Team

Team 15

Joe Wrieden

Benji Garment

Marcin Mleczko

Kingsley Edore

Abir Rizwanullah

Sal Ahmed

Assessment 2 Team

Team 12

Richard Liiv

Umer Fakher

William Walton

James Frost

Olly Wortley

Joe Cambridge

Risk Assessment and Mitigation v0.3

Introduction

Our Risk Management process consisted of first planning how we would carry out the next stages such as deciding the basic template we would use for our risk register. We then brainstormed as a group so as to voice everyone's concerns and systematically identified all possible risks stemming from various categories, from risks associated with time constraints and team organisation to the software itself. These were condensed, with risks with very low severities or likelihoods being eliminated, and refactored into a risk register as planned.

In the register, they were prioritised according to likelihood and severity, after we conducted qualitative analysis, since we bore in mind that this is a smaller project with negative ramifications not being critical. Thus we concluded that the level of detail did not need to be high either, and so did not focus on carrying out quantitative analysis, which may be less subjective but was too detailed and time consuming for our purposes. This prioritisation allowed us to put our focus on reducing risks we had determined to have problematic consequences. Our risks are tabulated with an unique ID for each one, the type/area, a description, likelihood, severity, mitigation and ownership. We felt all these columns are appropriate and necessary to fully be prepared in order to deal with any problems that may arise. We use three levels of likelihood and severity: low, moderate and high.

We split the risk types into People, Project, Product and Technology. People-type risks would be risks related to team members and the customer(s). Project-type risks would be risks that jeopardise the project as a whole. Product-type risks are risks that affect the outcome of the project, i.e. the product itself. Technology-type risks are to do with technical restrictions of the users and developers of the product.

The reason we chose those types of risks is because it categorises each of the risks we identified into appropriate groups that are easy to understand. This also helped us assign the most appropriate owners to the risks based on their skills.

Next, we planned to reduce the impact and probabilities of risks that required attention, by laying out avoidance strategies, mitigation strategies and contingency plans. Finally, we delegated the responsibility of monitoring certain risks to every individual in the group for the duration of the project. This consisted of reassessing the likelihoods and severities of risks under an individual's jurisdiction. Each person came back to the register regularly and updated the likelihood and severity of each risk they were responsible for and if they detected a great enough change, they would have to report the change in status and raise their concerns in our frequent meetings. New risks could also emerge and are added.

Our adherence to this carefully constructed process may have taken up greater portions of our time, but allowed for us to work well together as a team and provide deliverables of good quality on time as we had prepared for any risks that may have come beforehand.

Risk Register

ID	Type	Description	Likelihood	Severity	Mitigation	Owner
R1	People	Too many people working on the coding aspect and not enough people working on the documentation, or Vice Versa.	M	H	Split the team into half. Half of the team working on the code and the other working on the documentation. Ensure there is communication between each team so that any updates can be made clear. This is done via Discord.	~~Marcin Mleczko~~ **Will**
R2	People	Specific sections of documentation are not complete and specific requirements that were intended to be implemented are no longer implemented.	L	M	Have deadlines for specific sections of code/documentation to be completed for each week to ensure the team is on track and everything that was set is completed.	~~Sal Ahmed~~ **Umer**
R3	People	Team members not being sure of what tasks they will need to perform for that work, leading to a delay in completion of project/documentation.	M	M	Have a weekly to do list which provides each member of the task that they will need to complete during that week which will help with the workflow.	~~Sal Ahmed~~ **Umer**
R4	Project	Software Engineer becomes unavailable.	M	M	Involve a second Software Engineer to increase bus factor.	~~Joe Wrieden~~ **James**
R5	Product	Product is over-engineered, i.e. features that are not required are implemented, unnecessarily using up resources.	L	M	Enforce that the Engineers stick to the Architecture models as defined in the documentation.	~~Kingsley Edore~~ **Olly**
R6	Technology	User's computer has insufficient memory to deal with seven or more boats at a time.	M	H	Inspect program for optimisation opportunities each sprint, such as removing memory intensive operations from loops.	~~Joe Wrieden~~ **Richard**
R7	Technology	User's computer has an outdated or incompatible version of Java and is unable to execute the game reliably.	L	H	Write the program in a widely supported version of Java, especially by libGDX, such as Java 8.	~~Benji Garment~~ **James**

R8	Project	Initial schedule is greatly under-estimated and does not provide adequate time for the project.	L	H	Over-estimate length for each task to ensure that even if they are delayed, the deadline is not delayed too (i.e. the Cone of Uncertainty [1]).	~~Marcin Mleczko~~ **Will**
R9	Project	Requirements are introduced or altered by the client and must be implemented before the deadline.	M	M	Abide by agile methodologies, in this case Scrum, with short weekly sprints in order to address changes in requirements.	~~Abir Rizwanullah~~ **Joe**
R10	People	Requirements implemented are not to the stakeholders/customers liking.	H	H	Providing the customers with the week's prototype/deliverable in weekly meetings to discuss any of the requirements they feel may be at odds with the current version of the implementation (as in clarification of requirements).	~~Kingsley Edore~~ **Richard**
R11	Project	Members are not accountable for any faults during the project, leaving ambiguity as to who should correct them.	L	M	Enforce ownership of each task when they are delegated in each sprint.	~~Marcin Mleczko~~ **Olly**
R12	Project	Collaboration tools become unavailable for the team, thus limiting the productivity for a given time period.	L	M	Guarantee each member is contactable through more secure means such as email if necessary.	~~Kingsley Edore~~ **James**
R13	Product	Coders do not pay attention to the changing requirements.	L	M	Have a weekly meeting where the documentation team discusses what changes have been made to the requirements due to the team-customer meeting.	~~Joe Wrieden~~ **Olly**
R14	People	No one on the team has high enough coding skills to produce the required product.	L	H	Email the client with the problem. ~~Ask for an extended deadline.~~	~~Abir Rizwanullah~~ **Richard**
R15	Technology	Software Engineers are unsure on what kind of libraries can be extended from in order to implement the game.	M	H	Research is done beforehand into licencing and what libraries can be reused.	~~Benji Garment~~ **James**
R16	People	Someone on the team suddenly stops responding and ceases to complete the work that they were set.	M	M	Make sure there are enough people covering other members so they can finish the incomplete work.	~~Sal Ahmed~~ **Umer**

R17	Technology	Users monitor may not have a 1920 x 1080 quality resolution which may affect gameplay quality.	L	L	Provided a scalability option which allows the games to run smoothly regardless of resolution.	~~Benji Garment~~ **Will**
R18	Product	All requirements are not implemented within the product.	H	M	A documentation is provided to the customer/stakeholder with the requirements that are not implemented in the game with a justification as to why the requirement has not been implemented.	~~Abir Rizwanullah~~ **James**
R19	Product	Libraries we reuse may be unreliable and may affect our product quality.	M	H	Research is done beforehand into good quality libraries in order to pick the most suitable and reliable game library for the project.	~~Benji Garment~~ **Will**
R20	**Product**	**Leaving obsolete code when editing new requirements.**	**L**	**L**	**Double check the function of each segment of code**	**Joe**
R21	**Project**	**Not being able to adapt to the previous style of the project.**	**L**	**M**	**Analyse and familiarise oneself with the previous group's work.**	**Umer**
R22	**Product**	**Inadequate JUnit testing which may not catch errors in the code.**	**M**	**M**	**Have at least two people going through the tests to ensure the quality of the tests.**	**James**

References

[1] - McConnell, S. (2006). *Software Estimation: Demystifying the Black Art*. Microsoft Press

Bibliography

- Griffiths, M. *The Top Five Software Project Risks*. Project Smart. Retrieved 11 November, 2020, from https://www.projectsmart.co.uk/top-five-software-project-risks.php
- Robertson, M. (2020, July 08). *What are the 10 biggest risks in software development?* Codebots. Retrieved November 11, 2020, from https://codebots.com/library/way-of-working/what-are-the-10-biggest-risks-in-software-development