Introduction

The risk management process is an important process in the software engineering process. There are multiple steps to doing this: identification, analysis, evaluation, treatment and review [1].

Risk identification began with a discussion which identified risks based on past experience or were grounded in common sense. However, as a team, we do not have much experience with software engineering or game development so research was also a crucial aspect to identifying risk. This research focused on risk management within small teams or in game development, however we also considered broader cases. Identified risks were given an ID and a description in the risk register.

Risk analysis and evaluation was then carried out, where the risks were then sorted into 3 types: Project, Product and finally Project + Product. These categories specified whether the risks would affect the project, product or both and allowed for easier identification of risk mitigation strategies. The severity of the risk and the likelihood of it occurring was then assessed and this was also recorded in the risk register, where L shows low severity/likelihood, M shows medium severity/likelihood and H shows high severity/likelihood. By having too many options, the risk register can become too complicated, especially when there are not many risks. This was avoided by having 3 options for both the severity and likelihood which gave 9 possible overall options for the risk which was ideal for this scale of project.

Once the risks are identified and the severity and likelihood assessed, the next step in the process is to identify mitigation strategies to employ to minimise the effect of the risk.

The final part of the risk assessment is to continuously monitor and reassess risks in the risk register. As this task is too much for one person and because some risks are specific to certain parts of the project. The risks were assigned to an owner, who is responsible for checking the risk and its severity and likelihood. The owner is also responsible for communicating with the rest of the team if the risk happens.

Risk Register

ID	Type	Description	Likelihood	Severity	Mitigation	Owner
R1	Project	Software developer goes off sick	M	M	Have multiple people working on the code at once so it isn't reliant on one person being there.	Ivo
R2	Project	Someone needs to drop out of the group	M	M	Ensure all work is done somewhere where the rest of the team can access it. e.g. Github, Google Docs. Project plan is adapted accordingly to reassign tasks to remaining team members.	Carys
R3	Project + Product	Customer is inaccessible	L	H	Send multiple emails and after multiple unsuccessful attempts to contact the customer, talk to the module lead.	Haiqal
R4	Product	Game has bugs that have gone unnoticed	M	L	Test the game with multiple 3rd party people who didn't work on it.	Ivo
R5	Product	Game doesn't meet expectations of the customer	M	H	Ensure that the meeting with the customer produces a clear set of requirements that must be met and that the whole team is clear on what the expectations are. If there is anything that is unclear, send emails to the customer to clear this up.	Caner
R6	Project	Struggling to meet final deadline	M	H	Ensure that there is a clear project plan before starting. People who complete their tasks early can jump in and help with aspects that are running behind. Encourage team members to be upfront about what they need help with if they are struggling to get things	Carys

ID	Type	Description	Likelihood	Severity	Mitigation	Owner
					done.	
R7	Product + Project	Customer has surprise requirements that need to be implemented	H	M	Ensure that the customer meeting clearly defines what the customer wants. Ensure code can be easily adapted to suit new requirements.	Alex
R8	Project	Mismanagement of files	M	H	Ensure files aren't kept on one person's account and are stored on the shared google drive.	Shravani
R9	Project	Work is shared unequally	H	M	Assign tasks early on and ensure everyone has a clear idea of the overall project plan and what they are working on. Reallocate tasks as needed	Shravani
R10	Product	Assets aren't free use	M	L	Ensure assets can be replaced if they need to be, and that we know where to find alternative assets so that they can be replaced if necessary.	Owen
R11	Project + Product	Unclear requirements	M	M	Brainstorm questions before customer meetings to gain a clear picture of what they want. If requirements are still unclear then arrange a second meeting with the customer or send emails to ask for clarification on certain points.	Alex
R12	Project	Arguments between team members [2]	M	H	Encourage team members to talk it out if necessary, if necessary talk to one of the lecturers and have them supervise the discussion	Alex
R13	Product	Missing important requirement [2]	M	M	Requirements are reviewed as implementation begins. If additional requirements come up, ensure they	Haiqal

ID	Type	Description	Likelihood	Severity	Mitigation	Owner
					can be added to the table.	
R14	Product	Requirements overly specific [2] and so attempting to reach all the requirements overcomplicates the code	M	M	Requirements are reviewed as implementation begins and ensure that they are realistic with what is possible to implement	Caner

References

[1] M. Nogueira and R. J. Machado, “Importance of risk process in management software projects in small companies,” Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, vol. 439, pp. 358–365, 2014. doi:10.1007/978-3-662-44736-9_44

[2] M. Schmalz, A. Finn, and H. Taylor, “Risk Management in Video Game Development Projects,” 2014 47th Hawaii International Conference on System Sciences, Jan. 2014, doi: https://doi.org/10.1109/hicss.2014.534.