University of York Department of Computer Science
Engineering 1

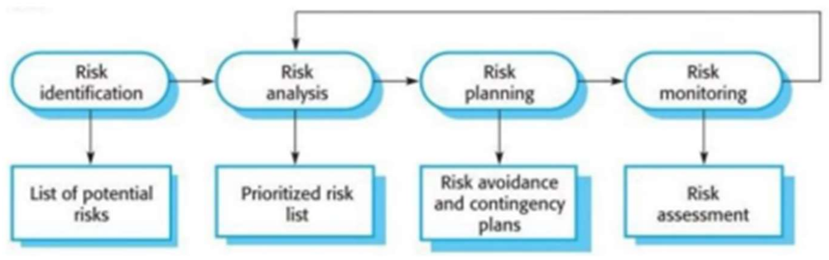
Risk Assessment and Mitigation

Cohort 3 Team 5 - alltheeb5t

Aaron Heald
Alex Gu
Arun Hill
Jade Stokes
Maksim Soshchin
Meg Tierney
Will Hall

Risk Management Process

Our risk management process involved 4 steps: risk identification, risk analysis, risk planning and risk monitoring.



[1]

The first step was to identify relevant risks that we might encounter during the project. In this phase it was important to eliminate any risks with negligible consequences or very low probability of occurrence. To help us identify risks, we used our past experiences working in team projects, both within the university and elsewhere. We also read through section 22.1 in *Software Engineering* [2] to get expert advice on risk management processes. To start we simply brainstormed some ideas, but then we began to categorise the risks into product related risks and human related risks.

We then began our risk analysis where we assigned each identified risk a severity and likelihood rating. Two team members were involved in this process and they each individually gave a ranking and the average score was taken.

After the analysis we had a good understanding of the known risks we might encounter during our project. We then had to plan avoidance and mitigation strategies. To help avoid risks, we made sure to not have any critical dependencies with a low bus factor. Some mitigation strategies we employed were:

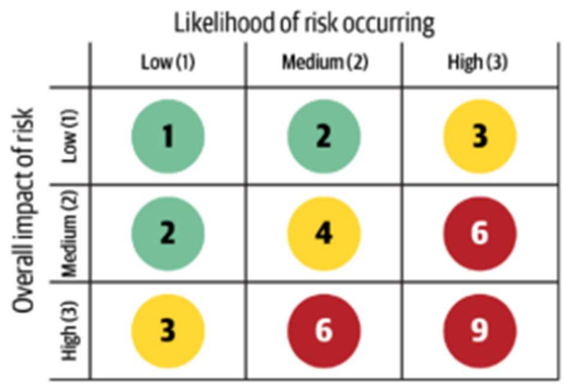
- Bi-weekly meetings to promote a close working environment
- Created documents for internal todo lists to keep everyone up-to-date on current tasks
- Cloud sharing to reduce the risk of important documents being lost.

As a group we also created contingency plans, such as using 3rd party sound and graphic assets if we run out of time or fail to make our own during the project timeline.

The final step of our risk management process is risk monitoring. This is an ongoing step throughout the project. We made sure that all team members were re-assessing risks related to their parts of the projects. It is the responsibility of the owner of a risk to make sure that the likelihood and severity of the risk is kept up to date.

Explanation of risk register columns

- ID: a unique risk ID so that the risk can be referenced easily elsewhere in the project
- Type:
  - Schedule - risk related to producing deliverables to deadline
  - Team - risk related to collaboration issues such as absent or ill teammates
  - Requirements - risk related to failing to produce a product and all its associated documentation that meets all requirements of the assessment, stakeholders and product brief
  - Code - risk related to bugs or poor code quality causing the product to fail to meet requirements
  - Tools - risk related to issues with third party libraries used in the game code
- Description: a short description of what the risk is.
- Likelihood: Low, Med or High. The probability of a risk occurring.
- Severity: Low, Med or High. How damaging a risk would be if it occurs.
- Risk Matrix (1-9): Represents the overall significance of a risk, making it much easier to understand the importance of a risk at a glance.



[3]

- Impact: an explanation of the impact this risk would have if it occurs.
- Mitigation: our planned strategy to avoid this risk and what to do if it occurs.
- Owner: the team member(s) responsible with tracking and updating this ris

Risk Register

ID	Type	Description	Likelihood	Severity	Risk Matrix	Impact	Mitigation	Owner
R1	Team	Sound designer becomes unavailable	Low	Med	2	Our plan is to design in-house sound for the game. If the sound designer becomes unavailable, the project may fail the UR_SOUND requirement.	As a contingency we researched 3rd party sound packs, so that we could implement them quickly if needed.	Jade
R2	Schedule	Estimated time taken to complete deliverables is underestimated	Med	High	6	This could result in the project being handed in late, or could reduce the quality of deliverables.	We made an internal deadline of one week before the official deadline for all tasks (apart from updating the website) so that we were aware of any delay well before the due date.	Meg
R3	Team	Member(s) of the team become ill or unavailable during the project	Low	Med	2	Losing a member of the team would increase workload on all other members and would require a rework of responsibilities.	We decided that if only 1-2 members became unavailable, we would be able to cope with the extra workload. If more members become ill we will ask for consideration from the teaching staff.	Arun, Maksim
R4	Tools	Faults in 3rd party code/graphics extensions	Med	Low	2	This could cause bugs in our code that we don't have the capabilities to fix, reducing our project quality.	If we find detrimental faults in a package, we will research and switch to a separate package.	Aaron, Arun
R5	Team	Conflict within the group (fall-outs, arguments)	Med	Low	2	This could hinder productivity for all members of the group, which could result in missed deadlines or unfinished tasks.	If 2 members of the team have an argument we will separate their responsibilities to keep them mostly apart. If members cannot stand to be on the team together then we can speak to the module staff to find a better resolution.	All members

R6	Code	Poor code quality	Med	Med	4	This could result in the game code being less maintainable, readable and extendable. This could also potentially result in the game running poorly and failing to meet NFR_SYSTEM_REQUIREMENTS	We decided that it's important that commits are monitored to keep a high standard of quality and maintain a higher bus factor.	Alex
R7	Requirem ents	Poor write-up quality	Med	High	6	This could cause the project to not make sense to markers, meaning we get a worse result.	Multiple members of each team will check over each deliverable, including a team member not majorly involved in that deliverable to avoid bias.	Maksim
R8	Code	Bugs in program	Med	Med	4	This would result in requirements not being met, which would lose marks. Severe bugs could make the game unplayable and not allow us to meet the UR_EXPERIENCE requirement.	Create automatic tests throughout development and rigorously test before any deadline.	Will, Alex
R9	Code	Issues with tracking and resolving merge conflicts in code	Med	Med	4	This could result in requirements not being met due to delays caused by complex or erroneous code merging.	Frequently make and merge small, incremental commits to the repository using continuous integration pipelines to ensure the smooth integration of different commits	Alex, Will, Aaron, Jade

References

[1] Figure 22.2 The risk management process

Sommerville, Ian. Software Engineering, Global Edition, Pearson Education, Limited, 2015. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/york-ebooks/detail.action?docID=5185655

[2] I. Sommerville, *Software engineering*. Boston: Pearson/Addison-Wesley, 2004.

[3] Figure 20-1. Matrix for determining architecture risk

Richards, Mark, and Neal Ford. Fundamentals of Software Architecture : An Engineering Approach, O'Reilly Media, Incorporated, 2020. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/york-ebooks/detail.action?docID=6029037

Created from york-ebooks on 2024-11-18.