Risk assessment and mitigation

Group 17

Team Loading

Joel Warren
Charlie Wilson
Jake Keast
Hari Bhandari
Ishrit Thakur
Joshua Hills

Risk management process

Risk Identification

In order to identify risks, we first conducted a work breakdown and listed the various tasks required for each section of the project. We did this first as it made identification easier and lowered the number of unidentified risks. This also allowed us to determine which sections of the project had the most risks, which also aided us in our monitoring and mitigation strategy. We then worked through each section of the project and identified risks by brainstorming as a group. We used brainstorming as it allowed us to consider multiple perspectives which ensured that we didn't miss any risks. Furthermore, by discussing the risks as a group we were able to challenge each other's suggestions and refine them, so that the risks identified were of a higher quality.

Risk Analysis

We then analysed the risks by assigning each one a likelihood and severity rating [fig.1]. We did this by group discussion and referring to literature to aid us in our decision. By consulting literature, we were able to consider the impacts of similar risks and appropriately allocate the likelihood and severity ratings. The likelihood allowed us to gauge the probability of a risk occurring, and the severity measured the impacts of each risk on the project. For both the likelihood and severity ratings, we used qualitative values from low to high as it made classification and prioritisation easier.

Risk Planning and Mitigation

We then developed avoidance strategies by brainstorming reasons why a particular risk would occur. By understanding the causes, we were able to find the most appropriate strategy for each risk. We then analysed the higher severity risks, and suggested ideas for mitigation by brainstorming and using online resources to aid us in our decisions. This allowed us to find proven methods that limit the exposure from such risks.

Risk Monitoring

We allocated owners for each risk to assess the likelihood and severity throughout the course of the project. We sorted the risks by the different sections of the project and assigned them to the corresponding members and ensured that there were at least 2 people monitoring each risk. This was done so that if a team member was absent or unavailable the risk would still be monitored. We also reviewed the risks at the end of each meeting to identify any changes that had to be made. This made sure that all members were aware of any changes to the risks or strategies, which ensured that the most suitable strategies were being used.

Presentation

The risks were documented using a risk register [fig.1], which also contained the likelihood and severity ratings as well as the owners of each risk. We included a risk type column to group similar risks, which made it easier to locate risks and identify which sections of the project were most vulnerable. We also organised the risks within each group by their severity and likelihood, this made it easier to prioritise our time with the risks that were most important.

Inheriting the risk register

Team ownership of the project has since changed, with our team (Team 17) assuming responsibility. Consequently, the risk register was updated to reflect this. First, we collectively analysed the previous team's risk management process and the risk register to thoroughly understand them respectively. This was vital because our team was now ultimately responsible for the project and therefore the risks involved too. We then updated the risk register to reflect the change in ownership, which involved:

- Removing redundant risks
- Risk analysis
  - Updating likelihood and severity ratings to reflect change in team ownership.
- Risk ownership
  - Updating risk ownership to reflect change in team ownership.
- Risk register presentation
  - Tabular format and layout was maintained, with each risks' ID updated to distinguish between previous and new risks, denoted by a preceding '1.' and '2.' respectively.

Following this, we then repeated the previously described risk management process to reflect new risks arising from assuming unfamiliar work/code. Furthermore, additional requirements were given by our customer which needed to be reflected in the risk register.

Following our risk management process, we then added the new risks to the risk register.

Finally, weekly risk monitoring was then carried out in accordance with the described risk management process, on the updated risk register.

Risk Register

ID	Type	Description	Likelihood	Severity	Mitigation	Owner
R1.01	Project	Someone becomes unavailable	M	M	Ensure we have at least 2 people for each section.	Everyone
R1.02	Product	Implementation doesn't meet the requirements of the stakeholder	M	H	Trace the system requirements directly from the user requirements, which are sourced from the stakeholder	Charlie, Jake, Hari
R1.03	Project	A delay in a task can cause delays in dependent tasks	M	M	Appropriately estimate the complexity of each section, and provide realistic deadlines	Everyone
R1.04	Project	Deadlines not completed on time according to our gantt chart. Knock on effect for dependent tasks.	M	H	Ensure deadlines are met. Add extra time on our timeline as a margin.	Joshua, Joel, Ishrit
R1.05	Product and Project	The implementation is not well documented	L	H	Agree upon documentation style that all coders must follow. Before pushing any code to Git it must be documented.	Charlie, Jake, Hari
R1.06	Product and Project	Different code uploaded by programmers don't work together.	M	H	Weekly code review	Charlie, Jake, Hari
R1.09	Project	There is less communication within the group, when working remotely	M	L	Hold regular voice calls and have text/voice channels for each section of the project	Everyone
R1.10	Project	Inaccurate project management results in us not knowing if we are behind schedule.	L	M	Assess progress with weekly gantt charts.	Everyone

R1.11	Technology	The requirements can't be met with the chosen framework	L	M	Check thoroughly before implementation, and have a backup option just in case.	Charlie, Jake, Hari
R1.12	Requirements	Requirements are poorly defined/extended which may lead to scope creep	L	H	Systematically capture all requirements through the use of interviews, group discussions, use cases and prototypes	Joel, Joshua, Ishrit
R1.13	Risk Management	Poor risk management results in unidentified risks which could jeopardise the project	L	M	Weekly risk monitoring	Joel, Joshua, Ishrit
R1.14	Product	Code has bugs and logical errors	M	M	Test frequently, follow coding best practices and peer review code	Charlie, Jake, Hari
R1.15	Product	Implementation of components that aren't as important, can extend the schedule and result in delays	L	L	Organise tasks by importance	Charlie, Jake, Hari
R1.16	Requirements	If a user requirement changes or is no longer needed, identifying every change that need to be made in the implementation can be difficult and may extend the schedule	M	L	Implement using the system requirements, and organise the requirements by corresponding user requirements	Joel, Joshua, Ishrit
R1.17	Project	Failure in identifying complex components of the project and allocating time appropriately.	L	M	Create a project breakdown and a gantt chart to aid in appropriate time allocation.	Joel, Joshua, Ishrit
R1.18	Project	People's	L	M	Conduct the project	Everyone

		assignments don’t match their strengths			breakdown as a group and let each member choose the tasks they want to do.	
R1.19	Project	Low productivity either individually or as a group	L	M	Effective project management using weekly gantt charts and effectively communicating any adjustments that need to be made	Everyone
R2.01	Product and project	Unexpected new requirements, adding to the time and complexity.	M	M	React accordingly by updating requirements, risk register. Extend time on implementation if necessary	Everyone
R2.02	Project	Unfamiliarity with assumed code	H	L	Take time to familiarise with code	Charlie, Jake, Hari
R2.03	Require ments	Implementation does not satisfy assessment 2 requirements	L	H	Work on requirements based on necessity (must / should etc)	Charlie, Jake, Hari
R2.04	Project	Unfamiliar with assumed deliverables	H	L	Take time to familiarise with previous deliverables	Joel, Ishrit, Joshua
R2.05	Product	Unexpected obstacles / inefficiencies in the code requiring longer than expected refactoring	M	M	Take the necessary time refactoring, extend time on implementation if necessary. Start working on implementation early to ensure deadlines are met.	Charlie, Jake, Hari
R2.06	Product	Controls unresponsive	L	H	Make sure that testing is thorough	Hari (and others)
R2.07	Product	Unenjoyable user experience - not for target audience	L	M	Give copy of game to people that fit the target demographic and collect feedback	Everyone

[Fig1]