

Risk assessment and mitigation

Group 21
Generic Games

Josh Thomas

Andrew Palombo

Oscar Gunn

Scarlet Desorgher

Immanuel Ghaly

Madeleine Nielsen

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. It 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.

ID	Type	Description	Likelihood	Severity	Mitigation	Owner
R6	Product and Project	Different code uploaded by programmers don't work together.	M	H	Weekly code review.	Immanuel, Oscar, Andrew
R5	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.	Immanuel, Oscar, Andrew
R12	Requirements	Requirements are poorly defined which may lead to scope creep	L	H	Systematically capture all requirements through the use of interviews, group discussions, use cases and prototypes.	Josh, Oscar
R16	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 the corresponding user requirements.	Josh, Oscar
R4	Project	Class diagrams and object diagrams are not finished on time according to our timeline. The implementation may not be finished on time	L	H	Ensure deadlines are met. Add extra time on our timeline as a margin	Scarlet, Andrew, Madeleine
R1	Project	Someone becomes unavailable	M	M	Ensure we have at least 2 people for each section.	All of us
R3	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	Scarlet, Madeleine
R18	Project	People's assignments don't match their strengths	M	M	Conduct the project breakdown as a group and let each member choose the tasks they want to do.	Madeleine, Scarlet
R17	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.	Madeleine, Scarlet

R8	Project	One of the programmers is unavailable	L	M	Involve the other two programmers. Split up the work between them.	Immanuel, Oscar, Andrew
R10	Project	Inaccurate project management results in us not knowing if we are behind schedule.	L	M	Assess progress with weekly gantt charts.	Scarlet, Madeleine
R9	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.	All of us
R7	Project	Different programmers working on the same requirement.	L	L	Create a schedule for the implementation. Assign different requirements to different programmers. Communication available between programmers.	Immanuel, Oscar, Andrew
R19	Project	Low productivity either individually or as a group.	L	L	Effective project management using weekly gantt charts and effectively communicating any adjustments that need to be made.	All
R2	Product	Implementation doesn't meet the requirements of the stakeholder	M	M	Trace the system requirements directly from the user requirements, which are sourced from the stakeholder	Immanuel, Oscar, Andrew
R14	Product	Code has bugs and logical errors	M	M	Test frequently, follow coding best practices and peer review code.	Immanuel, Oscar, Andrew
R15	Product	Implementation of components that aren't as important, can extend the schedule and result in delays	L	L	Organise tasks by importance	Immanuel, Oscar, Andrew
R11	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.	Immanuel, Oscar, Andrew
R13	Risk Management	Poor risk management results in unidentified risks which could jeopardise the project.	L	M	Review risks weekly	Josh, Immanuel

fig.1