

Risk Assessment and Mitigation

Group Number: 10

Team Name: Decassociation

Group Member Names:

Mohammad Abdullah

Tom Broadbent

Poppy Fynes

Owen Lister

Michael Marples

Lucy Walsh

Risk management and mitigation is essential in a project, regardless of the context, to ensure a high quality final product is produced. Consequently, we have followed a rigorous process for risk management and mitigation. This process consisted of risk identification, risk analysis, planning for the eventuality of the risks becoming reality and finally risk monitoring.

The risk identification step resulted in our risk register below, showing the risks we have identified, their description, impact, severity, likelihood, how we will mitigate the risk and their ownership (who is responsible for mitigation essentially), all in a systematic, tabular format. To identify these risks, we had to consider the different types of risks and how each type might affect the team and project in different ways. All risks fall into one of the following categories: project schedule and resources, product quality and testing, project requirements.

Project schedule risks affect the schedule and deadlines of the project, product quality and testing risks are based around the quality of the end product (Piazza Panic) and the testing process to ensure the quality of the end product, and project requirements risks focus on the project requirements and any changes to them (such as the requirements change between assessment 1 and assessment 2).

During the risk analysis process, we assigned a likelihood rating and a severity rating to each risk, with both categories having a rating scale from 'Low' to 'High', which allowed us to appropriately focus on risks, taking more time to account for risks with a higher severity rating. Risks with a 'High' severity rating are considered as nearly catastrophic.

The risk planning process was for mitigation and avoidance of risks. It's mainly about mitigation because we can't prevent most of the risks; it's about dealing with situations as they arise by making sure we have contingency plans. Therefore, we came up with a way to mitigate each risk we identified which also involved team members having ownership of certain risks based on the work they focused on the most. However, a few of the risks are 'owned' by the full team because for example if we lost one team member, we all need to be prepared to step in and take on a larger workload, it's everyone's responsibility. This ownership also links into our risk monitoring process and procedures.

Risk monitoring in our team is mainly done through the ownership of our identified risks. Every risk is 'owned' (and therefore monitored) by more than one team member so it's highly unlikely that a risk would go unchecked and that any problem(s) related to a risk would go unreported. To ensure that this is the case though, we meet weekly, discussing all progress made in the project, any issues that have arisen, and we also share a meeting history document which details all work that has been done in each weekly period, so any risks that are a real problem will be reported in this document for the full team to see (this can be viewed on our website: <https://decassociation.github.io/>).

Finally, to ensure that all the risks are relevant and that as many risks as possible are identified and mitigated, the risks are reviewed on a regular basis, as all documentation is reviewed to ensure coherence and a high quality.

<i>Risk</i>	<i>Type</i>	<i>Impact</i>	<i>Likelihood</i>	<i>Severity</i>	<i>Mitigation</i>	<i>Ownership</i>
R1 - Loss of One Team Member (for more than two weeks)	Project schedule and resources	If a team member were to no longer be present due to personal or health reasons, the rest of the team would have more workload to complete individually in the same time period.	Medium	Medium	Working on parts of the project in pairs and good documentation of code can mitigate the impact of a team member leaving. Some team members must be prepared to take on the workload of the lost person as well.	Entire team
R2 - Loss of More than One Team Member (for more than two weeks)	Project schedule and resources	If more than one team member was lost, we would likely struggle to keep up with the project workload at times. It would create a stressful situation and a rush to get everything done.	Low	High	All members must be prepared to step in and take on more work than they are assigned and have that work take priority over other work when necessary. We can't prevent team members from being ill or having personal issues so we must deal with the situation as best we can.	All remaining members of the group to ensure a high bus factor.
R3 - Loss of Data	Project schedule and resources	Documents, assets or code could be lost due to factors such as file corruption and file mismanagement. This would cause already completed work to have to be redone.	Medium	High	Upload the project to an online server to prevent data loss (e.g. github and Google Drive). Make backups of the project in case of data corruption.	Entire team. We all need to be careful of file protection.
R4 - Hardware Failure	Project schedule and resources AND product quality and testing	The hardware of team members could potentially fail, causing them to not be able to/have a more difficult time contributing to the project.	Low	Medium	Find alternative ways to team members to access the project data in the event of hardware failure.	Entire team
R5 - Bugs in Implementation Tools	Product quality and testing	Bugs and issues in the libraries used to implement the product could cause delays in production as the team takes more time to work around issues within the tools rather than focusing on implementation.	Medium	Low	Assess the tools used to create the project and determine if they are reliable enough to be used without problems.	Poppy, Mike, Lucy and Mo
R6 - Bugs in Testing Tools	Product quality and testing	Testing tools developed by the team could potentially be bugged, causing	Medium	Low	When developing tests, take extra care to ensure that the tests have been crafted correctly so that they are	Poppy, Mike, Lucy and Mo

		issues where problems present themselves later within the project that may take more time to fix at that point. This would affect productivity.			accurate.	
R7 - Misunderstanding of Requirements	Project requirements	The requirements given by either the brief or the stakeholder could be misunderstood, which could cause the team to lose progress if previously completed sections of the project have to be edited due to a misunderstanding.	Low	High	Read through and break down the requirements thoroughly to ensure that the entire team has a proper understanding of what is required from the project, and ensure there are at least 2 people who track the requirements and ensure they are up to date.	Owen and Tom
R8 - Misuse of Assets	Product quality and testing	Visual and audio assets taken from online sources could potentially be misused, which may breach the copyright policies offered by the asset's licence.	Low	Low	Ensure that when sourcing assets online that the licence of the asset is acknowledged and followed correctly. Additionally, avoid asset sites that may not be following copyright guidelines correctly.	Poppy, Mike, Lucy and Mo
R9 - Scope Increase	Project requirements	If features are added to the project, or less essential features are focussed on, the scope of the project could become too great to complete within the allotted time limit.	Medium	Medium	Prioritise the most essential requirements of the game and save the implementation of features deemed optional by the stakeholder until later in development.	Owen, Tom and Mo
R10 - Productivity Issues	Project schedule and resources	If not all members of the team contribute significantly or equally to the project it will likely increase the workload on the rest of the team or risk the project not being completed fully for the deadline.	Medium	Medium	Frequent meetings to check on each team member's progress can help understand who is contributing the most and least to the project, and actions can be taken using this information if needed.	Owen, Tom and Michael
R11 - Poor Code Design	Product quality and testing	Code that has been poorly designed/implemented is likely to break or be unusable further into the project, forcing the team to use more time fixing mistakes later on in development.	Medium	High	Have multiple team members review code to ensure that the approach being used to create certain features is sound. Research into better ways to create certain code modules if necessary.	Poppy, Mike, Lucy and Mo

R12 - Lack of testing	Product quality and testing	A lack of sufficient testing can cause flaws in both the game's code and the gameplay itself to be present. These could present themselves later on in the development time and cause issues that may not be fully fixable by that time.	Low	Medium	Ensure that the amount of tests and types of tests being performed is kept track of so that it is easier to notice which sections of the project have not been tested sufficiently.	Poppy, Mike, Lucy and Mo
R13 - Code/implementation doesn't match up with documentation	Product quality and testing AND project requirements	If the code for Piazza Panic doesn't match up with the documentation for the project (such as the architecture class diagrams and/or the requirements) then there will be a large incoherence. This may significantly impact how our implementation meets the project requirements and the quality of our end product.	Medium	High	There will always be at least 2 people who are up-to-date with all of the documentation and are responsible for highlighting any issues and inconsistencies between the code, requirements and architecture documentation mainly, but also the rest of the documentation where appropriate.	Owen and Tom