

# Changes to Risk Assessment

Mozzarella Bytes | Team 18

Assessment N°2

Daniel Benison

Elizabeth Hodges

Kathryn Dale

Ravinder Dosanjh

Callum Marsden

Emilien Bevierre

*Added Risks:* During the completion of our project, we regularly used our Risk Assessment to help us combat issues that we faced. We encountered three new risks that we added to our risk assessment and mitigation plan - R12 to R14. Team members regularly asked for help from others throughout the project and, once we added this risk to the assessment, all members were very good at following the mitigation strategy to minimise the impact of this on the project. Furthermore, we realised that we had yet to consider the risks associated with testing our project, hence the addition of R13. R14 was added as a result of the University holidays leading to us all being split across the country and so we had to consider how we'd manage this.

*Mitigation Changes:* Few changes have been made to the mitigation strategies in our risk assessment, as in general our thorough initial risk assessment meant that we were able to successfully navigate challenges as they arose. The mitigation for R1 has been altered, as we found that GitKraken was more suitable than a command line interface because it provides an interface that makes it easier to resolve merge conflicts. Some of the owners of each risk have been updated to reflect changes in the tasks assigned to each person. For example, as Kathryn designed the Gantt chart and took on a greater role within group communication, she took on R11.

*Likelihood and Severity Changes:* As a result of risk reviews, some changes have been made to likelihood and severity ratings for certain risks. The likelihood of R11 has been reduced to medium, because we have found that we were very adept at keeping up with what we were all doing.

We failed to fully follow the mitigation strategy from the risk assessment for R2. As many people were working on the code on different devices, leading to a high frequency of commits, we found that it wasn't practical to try and have a single up-to-date back up of our project, especially with every member having a local copy of the code that they were last working on. Although this was a high severity risk, the fact that the chances of GitHub going down permanently and losing our work is extremely low, lead to us making the decision that it was more time efficient not to make regular back-ups that quickly become redundant.

*Successfully Mitigated Risks:* The most significant risks that we encountered in this assessment were R4, R7 and R12. We dealt very successfully with R4 and R7 because, by keeping in regular email contact with the client, we were able to discuss developments to the project with them and allowed us to question them over issues that we had, as well as gave them an opportunity to highlight any changes that they had. Furthermore, it meant that we were able to present our ideas for the development of new features to the client to ensure that they would be suitable prior to implementation. This saved us a lot of time as it meant that we didn't go ahead and anything that the client didn't want.

R12 was very successfully mitigated. We kept in regular contact as a team and all members were quick to offer help to each other. This did much to help move the project forward and ensure that all elements were completed.

Link to Original Risk Assessment:

<https://emhodes.github.io/SEPR-game/assessment1/Risk1.pdf>