Engineering 1
Group Assessment 1
Risk Assessment and Mitigation Document

Cohort 2 Team 12:

Umer Fakher

William Walton

Richard Erasmus Liiv

Olly Wortley

James Frost

Joe Cambridge

Introduction to Risk Format and Level of Detail

The Low-Moderate-High scale is a simple and clear format for presenting risks applying to a small-scale project such as ours. For both the severity and likelihood of a risk, we have used this system since it is easy to understand and colour coded for even greater clarity.

We have classified each of the risks as one of three categories: product (regarding quality/completeness of the game), project (regarding the project resources and schedule) and business (regarding the organisation procuring/developing the software). These types were inspired by research into risk types in software engineering.

In terms of the level of detail, we have provided a coherent description of each risk which sets out what it entails and how it would affect the project and team, as well as how to mitigate it in the event of its occurrence.

We have given each risk an owner as recommended and a backup owner as this was a simple way of reducing the chance of a risk resulting in a problem.

Risks Table

Risks							
ID	**Type**	**Description**	**Likelihood**	**Severity**	**Mitigation**	**Owner**	**Backup Owner**
R_MIA	Project	Team member unavailable for remainder of project	L	H	Ensure knowledge and skills are shared between team members, and form subteams to work on tasks	Umer	Richard
R_GITHUB	Project	Github server goes down so we cannot access the repository.	L	H	Back up files in local repositories	Olly	Joe
R_REQUIREMENTS	Project/Product	Errors or changes in user requirements	L	H	Use agile methods and frequently review requirements and how we are meeting them, meet with client often	Will	James
R_PRODUCTIVITY	Project/Product	Overall productivity affected by external factors e.g. COVID-19 situation	M	M	Make sure work is spread evenly and work is done in teams or subteams with support on tasks offered from all team members and set achievable, sustainable timeframes	Olly	Will
R_OBSOLETE	Business	Software being used becomes obsolete or unusable	L	L	Have backup pieces of software which can be used on all work and ensure that they are ready to be used in this event.	James	Umer
R_ESTIMATION	Project	Incorrect estimate of project completion date resulting in	L	H	Review schedule and plan as a team, keep constant track of what team members are	Richard	Joe

		overrun.			working on currently and how much work is left to do. Make specific duration estimates for each task.		
R_CODE	Product	Poor quality code or clashes in code produced by different team members	L	M	Make sure code meets software and user requirements, and review often. Test code frequently and appoint reviewers for different sections. Use continuous integration of code and lots of documentation to ensure this does not occur.	Joe	Will
R_RISK	Project	Avoidance of or inadequate risk management.	L	M	Include risk in estimations and complete documentation on risks involved with project.	Umer	James
R_SCALABILITY	Product	Code does not cater to changing requirements and is not reusable by other team members.	M	M	Write simple and efficient code with clear documentation, frequently test and review code.	Richard	Olly

Bibliography

[1] I. Sommerville, Software Engineering, Pearson Education, 2008, pp. 74-98.
[2] K. Schwaber and J. Sutherland, The Scrum Guide™, 2017 pp. 1-19