Risk Assessment and Mitigation

Cohort 2 Group 4 - THADJAM
Abdul Fofanah
Matthew Holleran
Toby Watchorn
Arwen Minton
Daneena Roydean
Jessica McKerill
Henry Bambrough

Risk Management Process:

The team used a lightweight 4-step risk management process as the first version of the project is small-scale. This meant a simple, iterative review cycle kept the process efficient without any excessive documentation overhead and ensured that risks were continuously identified, assessed and monitored throughout development.

1. **Identification:** Risks were identified during weekly sprint planning meetings through team brainstorming and reviewing the previous sprint issues discovered.
2. **Analysis:** After being identified, each risk was assessed qualitatively using a Low/Medium/High scale for the likelihood and impact severity
3. **Planning/Mitigation:** For each risk, the team proposed practical actions to take to reduce the likelihood and impact of every risk identified
4. **Monitoring:** At the end of each sprint, the risk register was reviewed, and the impact/likelihood scores were altered depending on situation changes. Each risk was also assigned an owner responsible for tracking it and updating its status during reviews.

Risk Register Format Justification:

Formatting: The group used a tabular register to manage risks. The tables are grouped by risk type (technical, organisational, scheduling, quality) to help keep each one concise, informative and easy to maintain throughout the project. Each risk entry records what the risk is, likelihood, impact, mitigation and ownership.

Risk Register

Technical Risk ID:	Risk Name	Likelihood	Severity + Impact	Mitigation	Ownership
TR1	New Tools such as LibGDX and UML.	Medium	Medium - Likely that there will be small defects and reworks from	Coding members begin to practise and ‘play’ with the new tools from	**Developers** - learns new tools. **The Scrum Master** - creates

			misconfiguration. New tools could cause early uncertainty and slower delivery; however, this will be reduced over time as users get used to the new tools.	week1. This early Familiarisation will help ease uncertainty before The main coding begins.	practice sessions and watches progress. **Method Leads** - focus is documentation.
TR2	Integration Issues	High	Medium - This could trigger rework cycles and delay The next demo. It is likely that integration problems will convert into reworks (extra time and costs) and Also, schedule slips.	By adopting continuous integration with small and frequent merges to a shared branch, where every merge must compile. If we schedule a daily integration sprint, it will help catch Incompatibilities and mean fixes are easier.	**Developers** - manage code merging and testing. **Architects** - Make sure UML and design align with the codebase. **The Scrum Master** - schedules daily integration checks(sprints) and creates meetings for issue resolution.
TR3	Performan ce Limitations	Medium	Medium - This will force late optimisation or feature cuts. This could result in a worse user experience and missed targets.	Use the performance baseline and keep graphics light with incremental testing (performance) After each feature, we should prevent late surprises.	**Everyone** - review the work done after daily sprints. **Method Leads** - should update the plan accordingly. **Members** - should do work agreed at sprints and communicate delays.
TR4	Version Control conflicts	Medium	Medium - Time will be lost and potentially stall the team near deadlines.	Should only merge features after code review and resolution. This minimises the risks involved with integration.	**Scrum master** - defines what version to use. **Architects** - should be updating developers and should make sure documents show the changes.
TR5	Software bugs	High	Medium - Loses time trying to resolve any bugs, both syntax and logic. It can be very costly in time, especially when integrating small	Write maintainable code and use version control in Git so can use rollbacks if needed. Also, document all changes	**Everyone** - ensures all documented changes are kept up to date. **Anybody writing code** - ensure it is maintainable and has

			parts of code into the larger final product.	in Git.	updated documentation.
TR6	Lack of traceability	Medium	Medium - This would make it hard to verify tasks which fit the requirements.	Git commits and continuous updates will be crucial.	**Developers** will Ensure documentation, and they will keep the commits linked. **The Scrum Maste**r checks progress.
TR7	Lack of testing	High	High - Could lead to undetected bugs or unstable features. This will reduce the quality of features.	We should implement unit tests and run playtests after each sprint.	**Developers** handle the Testing and monitor the inclusion of tests.

Organisation and Team Risk ID	**Risk name**	**Likelihood**	**Severity + Impact**	**Mitigation**	**Ownership**
OTR1	Members do not show up to meetings	High	High - May delay decisions or cause tasks to become misaligned. This will force reworks and idle 'blocked' time. This would delay tasks and testing time. This may lead to schedule slippage as the live feedback loop can't work with poor coordination and missed touchpoints.	Using WhatsApp as a continuous communication channel. Each member should post the progress and any roadblocks. This should also be daily. This will maintain the coordination even if attendance drops. Tasks won't stall.	**Scrum Master -** structures meetings and ensures engagement. **Team members** update through WhatsApp and Google Docs. **Method Leads** - check sprints and records patterns in attendance.
OTR2	Communication Breakdown	Medium	Medium **-** Misunderstanding could lead to duplication work (or missed work).	Clear + continuous updates through WhatsApp and a clear outline of roles and updates in the documents will help prevent this.	**The Scrum Master** will update through WhatsApp. **Method Leads** will maintain the planning document.

OTR3	Uneven Workload	Medium	Medium - Some members could be overworked and others may end up idle, which will reduce quality and morale.	Communicate over WhatsApp if anyone feels there is a misbalance, and then they can rebalance Tasks via communication.	**The Scrum Master** will rebalance the sprint tasks. **Method Leads** will track all of the changes and each member should assist as needed.
OTR4	Limited Availability	High	High - Less available members will cause sprint goals to not be met.	Continuous communication on WhatsApp means we can plan ahead, prioritising core features.	**The Scrum Master -** manages Scheduling**. Team Members -** make absences known early.

Quality and Requirement Risk ID	**Risk**	**Likelihood**	**Impact and Severity**	**Mitigation**	**Ownership**
QRR1	Misinterpretation of the customer requirements	High	High - Features will not meet their expectations. This would also waste the developing effort.	Ensure that the requirements document is checked and understood by everyone.	**Everyone** should view the requirements document.
QRR2	The final product does not meet the requirements.	Medium	High - Final product will not meet expectations or expected quality. This means the brief will not be met.	Ensure the team is on track to meet the deadlines and up to date with the work set. If anyone is behind, then ensure there are plans to keep them up to date via WhatsApp.	**Everyone** ensures they are on track. **The Scrum Master** can ensure people are meeting their deadlines.

Estimation	**Risk**	**Likelihood**	**Impact and Severity**	**Mitigation**	**Ownership**
ES1	Not meeting deadlines	High	High - Underestimating the amount of time a project will take to complete. Could lead to little time at the end of	Communicate over WhatsApp, and if anyone feels like they are not going to make a deadline, then let other	**Everyone** ensures they are on track with deadlines. **The Scrum**

			the project to complete everything.	team members know so they can help meet the deadline. Worst-case scenario: move the deadline (not ideal).	**Master** manages scheduling.
ES2	**Overestimating Capabilities**	Medium	High - would mean that users set themselves too much work or unrealistic standards to meet.	Everyone checks over each other's work, and if someone is struggling, they communicate honestly and as soon as possible to take action.	**Everyone** ensures their goals and work ethic are realistic and achievable.
ES3	**The team runs out of time for the project**	Medium	High - this would mean incomplete work and product briefs not being met.	Setting deadlines and completing work in small chunks. Weekly progress meetings to ensure we are on track.	**Everyone** attends weekly progress meetings and keeps up to date with their work.