Risk Mitigation 2

Cohort 2 Team 5

Harry Beaumont-Smith
Tom Nolan
Will Punt
Ruth Russell
Mimi Shorthouse
Lottie Silverton
Stanley Thompson

This document shows our approach to risk management for our software development project. Our approach was influenced by Somerville's process to risk management, which includes a four stage process: Risk identification, Risk analysis, Risk planning and Risk monitoring. This iterative process aligns with agile development practices and is appropriate for the scale of our projects scope and complexity.

Risk planning began by identifying all the foreseeable risks that may affect our project. This process began through team meetings and reviewing the interview and brief given from the client to create a clear plan of user requirements (requirements tables) with priorities and dependencies, which let us make development decisions which were more likely to deliver all the key user requirements reducing the risk of not completing the project before the deadline. For risk monitoring we utilised the approach of regular risk review at the end of our group meetings allowing for team members to discuss the status of the risks they were managing or raise concern for new potential risks that were not already identified. This was possible due to the small scope of the project and the low complexity of all the required systems.

We made a risk register to keep a record of any risks we have already identified and their appropriate management strategy. Every risk goes under analysis and is assigned a specific type, likelihood, severity, strategy and an owner. Each risk is first split into one of four categories: project, product, technology and business where project risks are risks concerning team management, planning and scheduling such as a team member falling ill for a significant amount of time. Product risks concern the possibility of faults in the final product produced such as the game's controls not being understandable or bugs found during merging different systems together in development. Technology risks are failures in the software or hardware that is needed to develop and maintain the game such as the risks of using libraries that might become unavailable during the development process, tools and infrastructure. Business risks affect the whole project's completeness such as change in management or the project becoming obsolete before completion.

Then all identified risks are assessed on the probability of the risk occurring and the potential impact this risk has on the project's objectives and duration. Within the risk register, Likelihood and severity are colour coded to clarify the overall impact of the risk to emphasise the importance of reviewing its risk status and we organised the risks with the highest impact first, making the most significant threats visible primarily to support decision making within our team and to scale the necessary management strategy to said risk.

For risk planning, for each identified risk we develop a specific management strategy plan combining both mitigation and contingency plans. Our plans ensure that we can effectively respond to any foreseen threats without compromising our project objectives and duration. Owners were assigned to risks to help keep track of risk status more easily as it divides the task of risk monitoring amongst the team, saving more time for development and implementation of the game.

Risk Register

This risk register represents our identified risks which are sorted via type and impact to our project deliverables. Each row contains a management strategy and an assigned owner to the risk.

ID	Type	Description	Likelihood	Severity	Management Strategy	Owner
R1	Technology	Physics step ties to framelate, leading to inconsistent gameplay	M	H	Delta-time audits	Tom
R2	Technology	The final game is incompatible with clients hardware	L	H	Software to be tested on multiple team members laptops to ensure functionality	Ruth
R3	Technology	IS/driver update breaks graphics	L	H	Freeze image rollback plan	Ruth
R4	Technology	University services fail (VLE, WiFi, local networks)	L	H	Ensure the project is non-reliant on any university hardware and local copies are frequently available	Mimi
R5	Technology	Plugin/library deprecated mid-project	M	M	Choose supported libraries fallback plan	Harry
R6	Technology	AntiVirus flags unsigned executable	M	M	Zip build publish checksums alternative host	Harry
R7	Technology	Engine upgrade breaks shaders/pipelines	M	M	Pin versions controlled upgrade window	Harry
R8	Project	New to Git, leading to accidental history loss	H	H	Create branches protect main frequent small commits	Stan
R9	Project	Time management conflict between team's other course modules	H	H	Ensuring all work is split evenly between team members and no individual team member is overwhelmed	Will
R10	Project	No clear backup of finished build	M	H	Store last 3 builds in hot and cold storage	Stan
R11	Project	last - minute feature add breaks stability	M	H	Rollback to previous version re-evaluate the code	Will
R12	Project	Inaccurate or ambiguous requirements gathered.	M	H	Thorough user requirements stage, performed by the whole team, with analysis	Stan

ID	Type	Description	Likelihood	Severity	Management Strategy	Owner
					donor interview transcripts	
R13	Project	Development team member becomes unavailable	L	H	All tasks can be assigned if a member is not available to complete their task	Tom
R14	Project	Understanding time for balancing	H	M	Separate balancing track extra playtests	Tom
R15	Project	Under- documented code becomes unmaintainable	H	M	Comment template short READMEs per system	Will
R16	Project	Poor team collaboration and miscommunication	M	M	Ensure that a clear outline of individual roles is set out, with each member having a clear idea of their own work and others	Harry
R17	Project	Relying on beta engine features	M	M	Prefer LTS/released features fallback plan	Will
R18	Project	Conflicts between team members in potential decisions	M	M	Having a clear method of resolving any issues and conflicts	Tom
R19	Project	Late joiner slows team due to onboarding gap	M	L	Onboarding doc buddy system	Lottie
R20	Product	Bugs	M	H	Game is tested out to make sure there are little to no bugs	Mimi
R21	Product	Website becomes inaccessible	L	H	Website is regularly maintained and tester	Harry
R22	Product	Inaccurate scoring system	L	H	Simple scoring rules keep values in a data file for instant hotfixes	Ruth
R23	Product	Rooms feel empty due to content shortage	M	M	Procedural decorators. Reuse patterns with tweaks.	Stan
R24	Product	Too many features added beyond initial scope	L	M	Clear method selection and planning to ensure all deliverables are reasonable within the time we have	Mimi
R25	Product	Gameplay unclear or objective not obvious	L	M	Clear tutorial screen implemented giving information about the game.	Mimi
R28	Business	Final delivery package rejected	M	H	Delivery checklist pre-acceptance dry run	Tom

ID	Type	Description	Likelihood	Severity	Management Strategy	Owner
R29	Business	External showcase build leaks unapproved content	L	H	Redacted build profile content flags	Stan
R30	Business	Inaccurate feature list in hand-in docs	M	M	Doc freeze verification pass vs build	Will
R31	Business	Stakeholder wants analytics report for showcase but data absence	M	M	Log minimal events now build basic dashboard	Lottie
R32	Business	Using music from ‘free’ site without proof	M	H	Keep licences/screenshots prefer CC0/library	Lottie
R33	Business	Using copyrighted assets	L	M	Check copyright rules before using the assets	Lottie
R34	Business	Music/SFX license gaps or expirations	L	M	Minimize/anonymize consent policy review	Ruth