

Risk Management

5a

The process our team decided to follow mirrors that which is laid in chapter 22 of Ian Sommerville's "Software Engineering". We decided on this after some research into various alternatives and came to the conclusion that this was the most easily comprehensible and covered all of the potential issues faced during our game's development window. The methodology is also simple enough for the context our project, as the development of our game is a relatively small endeavour, we did not want to waste time in dealing with large documentation overhead central to critical software, or implement processes intended for use with larger projects seen in books like "Managing Project Risk and Uncertainty" by C.B Chapman.

Sommerville's process begins by laying out the three different types of risks, characterised by what each risk affects. These include:

- Project risks, which affects the schedule or resources of a project
- Product risks, which affects the quality of the final product
- Business, which affects the organisation developing the software

Next an outline of the risk management process itself is identified, the 4 steps included here are:

- Risk identification, shed light on the possible risks
- Risk analysis, determine the likelihood of each risk, and the consequences of each
- Risk planning, strategies for mitigating each risk
- Risk monitoring, regularly assess and revise plans for risk mitigation

As a starting point for risk identification, our team brainstormed and looked for risks in 6 different categories: Estimation, Organisational, People, Requirements, Technology and Tools. This table can be found on our website and shows the results of said brainstorming. In the next stage, risk analysis we assessed the likelihood of each risk along with the severity, this table can also be found on our website. Following on from this mitigation strategies were thought of to counter each of these risks, the different strategies used here can be characterised by:

- Avoidance, try to reduce the likelihood of the risk occurring
- Minimization, try to reduce the impact of the risk
- Contingency plans, if the worst occurs, what is our back plan

Again this table can be found on our website. For the final stage in the process, risk monitoring, during our weekly meetings, each point in our risk register (described in the next paragraph) was discussed and evaluated on the basis of if it was more or less likely to occur, and if the consequences had changed for said risk.

Finally our team produced a risk register, the columns for this being ID, Type, Description, Likelihood, Severity, Mitigation and Owner. We believed that this was the most clear and concise way of representing the risk within our project, the risk register is shown on the next page.

5b

ID	Type	Description	Likelihood	Severity	Mitigation	Owner
R1	Project	Key team members are unavailable	Moderate	Serious	Reduce bus factor by ensuring a skill is not isolated to a team member	Whole team
R2	Project	Changes to the requirements	Low	Tolerable	Design with changing requirements in mind	Documentation team
R3	Project	Software is not finished before deadline	Moderate	Serious	Drop anything not essential, have more people dedicated to code	Whole team
R4	Project	Scope creep	Moderate	Serious	Links with R3, focus on core deliverables and features	Whole team
R5	Business	Code is hard to decipher for other teams	Low	Serious	Construction phase focuses on good practises, regular refactoring	Development team
R6	Product	The game is either too hard or too easy.	Low	Catastrophic	Constant play testing to ensure difficulty remains appropriate	Stakeholders/Users
R7	Product	Poor performance	Moderate	Serious	Optimise where possible, do not reinvent the wheel for things done better before	Stakeholders/Users
R8	Project	Code is lost/damaged	Low	Catastrophic	Make frequent pushes to the repository. Don't code too much before making backups to	Development team

					minimise potential losses	
R9	Business	Disputes between team members about how to proceed etc.	Low	Low-Moderate (depends on severity of the dispute)	Encourage polite communication, other team members should act as mediators to quickly resolve the issue	Whole team
R10	Product	Documentation is not on track to be completed before the deadline	Moderate	Moderate	Spend more time on documentation, move team members from code to implementation. It isn't as technical so it will be possible	Documentation team
R11	Product	Documentation is not high quality	Moderate	Low	Periodically make revisions to the documentation and have all team members review progress to ensure everyone is happy with it.	Documentation team
R12	Product	Website hosting issues	Low	Serious	Choose a reliable provider for hosting the website. Check periodically that the website is still up.	Website manager
R13	Project	Practicals are affected by ongoing strikes	Moderate	Low	Schedule alternative meetings so that project progress is not affected.	Whole team
R14	Business	Unexpected withdrawal of a	Low	Serious	Redistribute that team	Whole team

		team member			members work among remaining team members. Should still be possible to deliver goals without this person.	
R15	Product	Merging issues in Github	Moderate	Moderate	Practise good coding habits and always make requests before pushing anything.	Development team
R16	Business	Team members not attending meetings	High	Low	Good communication and making sure work is still allocated and done	Whole team
R17	Project	Two people working on the same thing - inefficiency and potential disparity in the code	Moderate	High	Create a good plan and follow it, regular meetings and communication between the development team to avoid this.	Development team
R18	Project	Going over budget (time)	Low	Catastrophic	Keep an eye on progress towards the end of the project and be especially mindful of the deadline.	Whole team
R19	Product	Unclear/Incomplete requirements	Moderate	Low	Continually review and update requirements as more is understood about the project and the limitations of development	Documentation team.

Bibliography:

Sommerville, I. (2004) *Software engineering*. Boston: Pearson/Addison-Wesley.

Chapman, C. (2002) *Managing project risk and uncertainty: A constructively approach to simple decision making*. Chichester: Wiley.