

Risk assessment and mitigation: Teams 23 & 27

Tom Rosewell

Sam Redgewell

Tom Keeler

Jake Stogden

John Limb

Adham Nouredin

Part A: Risk Format and Level of Detail

Our team identified three variations of risk that could possibly emerge:

- **Project** – affects project schedule or resources
- **Product** – affects product quality/completeness
- **Business** – affects the organisation procuring/developing the software

In order to account for these, our team delineated a four-step process in order to ensure that no risk could dramatically affect our project.

1. Identification of all conceivable risks
2. Analysis of each risks likelihood and severity
3. How each risk can be avoided, minimised, or mitigated

→ An ongoing monitoring of whether any risks are taking place

Before any programming tasks were assigned, our team thought it would be logical to have a brainstorming session, wherein we suggest all risks that could possibly emerge during the project. Based upon our research, [1] we decided to characterise each risk along 6 dimensions.

- **Risk ID** – lends itself for easy and accurate reference during discussion.
- **Risk Description** – a description of the risk.
- **Risk Likelihood** – the probability of the risk occurring: low, medium, or high.
- **Risk Severity** – the damage that the risk will afflict to the project if it occurs: low, medium, or high.
- **Risk Mitigation** – how our group will endeavour to prevent, mitigate, or minimise a given risk.
- **Risk Owner** – who is assigned to keep track of the risk and alert the team should it emerge.

Each of these makes up a column within a risk register, sorted by the most probable and most severe risks, to the least probable and least severe risks. There is a risk register for each of the aforementioned variations of risk: project, product and business. Each risk register has been coloured differently in order to allow for easier distinction between them.

As our project is a relatively small one, developing non-critical software, we have limited the scale of risk to between 1 and 10 for the severity, and 1 to 5 for the likelihood.

Part B: Tabular Representation of Risks

Project Risks:

ID	Description	Likelihood	Severity	Mitigation	Owner
R1	A team member falls ill and is unable to work for a period of time	2	6	Communication should be done through chats which everyone can access in case someone needs to take over another's work	Everyone
R2	A team member does not complete their task by the internal deadline set	4	7	Assess the reason behind the missed deadline. Distribute work among other team members to ensure development does not slow	Everyone
R2	A team member drops out or is unable to/won't assist in further development	1	9	Following from R1, ensure all work is well communicated so that other members can pick up where they left off.	Everyone
R3	Code is illegible to other developers (both in and out of the development team)	3	5	Have set conventions and standards established at the start of development. This can also be explained during meetings or through scrums.	Adham, Tom, Jake

Product Risks:

ID	Description	Likelihood	Severity	Mitigation	Owner
R4	The product is faulty	2	10	Ensure the code is subject to rigorous testing before production	Everyone
R5	Player's screen size is larger than the max tested for	3	2	Ensure throughout the development stage that the game screen can be resized and it scales proportionately	Josh Q
R6	Game is over-engineered with features not required	3	2	Have regular team meetings and outline exact requirements ensuring they are stuck to	Everyone
R7	Player's computer does not have relevant software installed to support the game	1	10	Ensure we use the highest supported JDK	Faris A, Josh Q, Louis H, Adham, Tom R, Jake
R8	Features which are required are not implemented	2	9	Prioritise developing the core features in the planning states, with a focus on functionality first	Faris A
R9	Player's hardware cannot support the game	1	8	Minimise intensive operations which may cause some machines to slow down	Faris A, Josh Q, Louis H
R10	The hard drive where the software is stored fails	1	9	Make use of online repositories such as GitHub to ensure the code can be accessed by any team member	Everyone
R11	The game is too difficult	2	5	Have people not on the development team sporadically test the game so it can be fine tuned	Everyone

Business Risks:

ID	Description	Likelihood	Severity	Mitigation	Owner
R12	New requirements are introduced by the customer mid production	3	5	Follow an agile approach to the development so that new requirements can be easily implemented in a short period of time	Everyone
R13	Goals are not met within the timescale planned	2	8	Have weekly team meetings and set deadlines so that the team knows at what stage they are at and whether focus needs to be directed elsewhere	Everyone

Bibliography:

- [1] <https://www.castsoftware.com/research-labs/software-development-risk-management-plan-with-examples>