UNIVERSITY OF YORK

DEPARTMENT OF COMPUTER SCIENCE

Risk Assessment & Mitigation

Cohort 2 - Group 17 (Rich Tea-m 17)

Group Members:

George Jopson
Ben Slater
Meg Tierney
William Potts
Jamie Burgess
Seyi Towolawi
Alex Staicu

The risk management process is split up into four main sections: risk identification; risk analysis; risk planning and risk monitoring. These steps were followed both in the risk mitigation stage of the original greenfield project, and in the subsequent brownfield development.

Risk Identification

This stage is concerned with identifying all potential risks that could pose a major threat to the software engineering process, the software being developed, or the development organisation [1]. Exploring common risks to a software development project was the first step to identifying risks. This research then allowed us to adapt those risks to our own project. The team members assigned to this deliverable brainstormed possible risks together and they were added to the risk register. Each risk was given an ID so they can be referenced by other risks and documents. We also considered past experience in group projects for risks that almost or did occur. We looked into different categories of risks and added a column in the risk register to record it. This was to assist with organisation and analysis/planning as there can be similar mitigation strategies for risks in the same category.

Risk Analysis

Each discovered risk is considered and a judgement is made about its likelihood and impact. This relies on personal judgement and previous experience so there is no correct answer it is just an estimation of the priority of the risk. A description of the impact was added as a column in the risk register to make it easier to complete risk planning. The impact level and probability level were given a rating from 1 (low) to 5 (high) as it is fairly simple and allows the use of a risk matrix to calculate priority. Priority level is the impact level multiplied by the probability level which results in four categories: very low (green); low(yellow); medium (orange) and high (red). Once all risks were analysed and given a priority, those with lower priority due to having very low probability or only minor consequences, were removed.

Risk Planning

Once risks have been identified and analysed, strategies must be put into place to ensure these major risks do not threaten the project. These mitigation strategies can include: avoidance strategies, which aim to reduce the likelihood of a risk occurring; minimisation strategies, which reduce the impact of a risk and contingency plans, to deal with a risk when it arises. To develop these strategies, we considered past experience of what did and didn't work when a risk arose. We considered what information should be collected throughout the project to uncover risks before they become serious. A mitigation strategy column was added to the risk register so when they occur appropriate strategies can be implemented.

The owner of a risk is the individual who is ultimately accountable for ensuring the risk is managed appropriately [2]. To decide who should be the owner we considered who the risk will affect the most and who has the best ability to be able to prevent/manage it. To manage these owners an Owner column was added to the risk register.

Risk Monitoring

In order to appropriately monitor and review risks we encouraged team members to report new potential risks or risks that have changed throughout the project. We created an online form so team members can express concerns and opinions without revealing their identities. Identified risks were reassessed regularly in group meetings where each risk is considered and discussed individually. A risk reassessment is performed regularly at all stages in the project to ensure assumptions about the product, project and business risks have not changed. A risk reassessment column was then added to the risk registry to keep a track of how often they should be checked. The likelihood and effects of a risk are also subject to change as more information about a risk becomes available and mitigation plans should be revised if necessary.

Priority risk matrix:
X = Impact level, Y = Probability level

5	5	10	15	20	25
4	4	8	12	16	20
3	3	6	9	12	15
2	2	4	6	8	10
1	1	2	3	4	5
	1	2	3	4	5

Risk Register [3]:

ID	Risk class	Risk description	Impact description	Impact Level	Probability Level	Priority Level	Prevention/Mitigation Strategy	Owner	Reassessment Date
1	Project - People	(Concentration risk) Dependency on a single key team member for a critical task	Delay in task completion, potential project failure	5	3	15	Documentation of critical processes. Each critical task will have more than one person overseeing/contributing to it to avoid having any single point (person) of failure. Conduct regular knowledge-sharing sessions, ensure backups are in place for key roles.	Meg	Biweekly
2	Project - Requirements	(Scope Creep) Continuous addition of new features beyond the initial scope of the project.	(Resource drain). The effort would be disproportionate to the marks given for the particular task and would be considered a waste of time and resources.	2	5	10	To make sure the team is not adding or changing features that shouldn't be added or changed. (Change control process). Document all the necessary features that are needed to be added and changes that are requested by the client.	Meg	Weekly
3	Project – People	(Communication failure) There has been a conflict between group members and communication has broken down.	Decreased productivity, team morale issues and dysfunctionality between team members	3	2	6	(Mitigation strategy for ID:1) Establish conflict resolution protocols, encourage open communication channels. Conduct regular team check-ins, address conflicts promptly	George	Weekly

4	Project – People	(Communication failure) Lack of communication causes multiple team members to do the same work	There will be multiple versions of the same work which will need combining/choosing between in a fair way to ensure everyone participates equally. Alongside delayed project timelines	3	3	9	Implement task tracking system, promote regular progress updates. We must establish clear task assignments, and encourage communication between members.	Will	Weekly
5	Project – People	Lack of communication causes a team member to do too much of the remaining work	There has not been equal participation and there is not enough remaining work to make it even.	4	3	12	Have regular workload assessments and promote open communication regarding task allocation. Conduct regular check-ins on workload distribution, provide support for overwhelmed	Georg e	Weekly
6	Project – People	A team member becomes temporarily absent for a specified period of time.	Delay in task completion and redistribution of workload	4	2	8	Document handover procedures and ensure clear task delegation. Cross-training of team members across various tasks could be the contingency plan, however the risk scales with team size, therefore evaluating the work ethic of each and every member should be feasible. Establish contingency plans for temporary absences.	Meg	As required (during the time of absence)
7	Project – People	A team member becomes temporarily absent for an unspecified period of time	Uncertainty in task completion, increased workload for remaining team members and decrease in team morale	5	3	15	Regular check-ins with absent team member(s), distribute workload among remaining team members. Also (Mitigation strategy for ID:8)	Seyi	Weekly, until return of absent member is confirmed
8	Project - People	A team member permanently drops out	We would only have 5 people which is may put pressure on the rest of the team due to increased workload. Also (Impact description for ID:8, 9)	5	3	15	Establish contingency plans for permanent drop outs of a single team member and consult module leader. Also, (Mitigation strategy for ID:7)	Seyi	When and if it happens.
9	Project – People	2+ team members permanently drop out	We would only have 4 people which is not considered enough to complete the project Also (Impact description for ID:8, 9)	5	2	10	If some deliverables are dropped, the remaining team will work on the new, reduced deliverables. If all are dropped, consult module leader. Also, (Mitigation strategy for ID:7 & 10)	Seyi	When and if it happens.

10	Project-People	A team member has been assigned too much work and reports they will be unable to complete the work on time	Delay in task completion.	5	4	20	Regular Evaluation and distribution of the workload according to the skillset of the members. Cross-training of team members across various tasks so that no concentration risk takes place.	George	Weekly
11	Project - People	A team member hasn't completed their work by the deadline and didn't report it	Project delays, compromised task dependencies	3	2	6	Clear reporting protocols, task tracking systems and establish reporting expectations in order to follow up on missed deadlines promptly	Seyi	Weekly (until its no longer a problem)
12	Product and project - Requirements	There has been a drastic change in requirements	Increased workload, potential delays, scope creep	5	3	15	Robust change control process and immediate impact assessment. Also, negotiate changes with stakeholders.	Jamie	As needed, based on frequency of changes in requirement
13	Product and project - Requirements	The change in requirements increases the workload by a large amount but the deadlines are not pushed back	Overworked team, Increased workload, compromises in terms of quality (lackluster product)	5	3	15	Negotiate deadline adjustments and time extensions. Also assess the resource reallocation and workload.	Will	Weekly (until it's no longer a problem)
14	Product – Technology	Inadequate testing leads to issues with the product	Lackluster product and customer dissatisfaction	5	2	10	Comprehensive testing strategy alongside an intricate benchmark for quality needs to be established. Also surveys to incorporate feedback.	Jamie	Monthly
15	Product -Tool	A tool relied on for a large portion of the project becomes unusable.	A new tool must be found and code rewritten which will require extra resources and increase workload.	4	2	8	Research all tools used extensively to ensure they seem reliable for the foreseeable future. Research alternates so they can be quickly implemented if necessary.	Alex	Biweekly
16	Product – Technology	When changes are made to the code, it works on the changer's device but not on different hardware/software.	This could mean not all members of the team can run the code and develop it. It also means when the game is being marked, the module leaders cannot run the code.	4	2	8	Code should be tested on multiple different types of devices at every major change in the code to ensure it can be run on different devices. If it cannot, we can use github to revert to a previous commit.	Alex	Weekly
17	Product - Technology	When merging two branches there are merge conflicts or similar issues	This means the current version of the code will not run/is incorrect or new code cannot be added to the main branch.	4	4	16	Analyse the code and git log to see how this has occurred and resolve any merge conflicts. If unsure, revert to last commit.	Alex	Weekly

18	Project – Estimation	Time to implement all requirements is underestimated, perhaps due to initially misunderstanding requirements.	This will lead to a rushed end of development, putting undue pressure on the team. This rush will also likely cause a drop in standards, sacrificing the quality of the final product.	4	3	12	Have a list of high-effort low-impact changes to the project. These can be dropped in the case of an under-estimation of how long it will take to do work. Therefore, the development team has time to focus on the important work. Also regularly review work left at each team meeting, to re-evaluate any estimates with gained knowledge. If it seems like we under-estimated the time taken to complete a section of work, that work can be shifted to the "buffer" week of time planned at the end of the project.	Ben	Weekly
19	Project - Estimation	Amount of effort needed to complete different aspects of the project are incorrectly estimated.	This will lead to some team members getting overloaded with work while others having nothing to do.	3	3	9	Agilely respond to new information about team members workloads, by moving work to team members who managed to complete their work quickly.	Ben	Weekly
20	Project - Tool	IDE is slow/inefficient to use, or not suited to the team's workflow.	This would lead to the team's development slowing down due to issues dealing with the IDE. This could lead to all implementation taking significantly longer.	4	2	8	Have back-up IDEs (such as Visual Studio Code) that members of the team are already familiar with, that can be switched to if the initial IDE (IntelliJ) is causing problems.	Alex	Weekly
21	Project - Technology	Team misunderstands legacy code (perhaps due to poor documentation from previous team)	This can lead to initial features being developed in a way that could break already implemented systems. Or new systems could be made in a way that doesn't interact well with legacy systems.	4	4	16	Have time planned in to initial development to account for mistakes made in taking over code. Then version control systems can be used to roll back ill-advised new feature implementations.	Seyi	Weekly for first 2 weeks of project while team is still understanding the code base
22	Project - Requirements	Previous team missed major requirements from assessment 1.	This will lead to extra work implementing these original requirements.	4	2	8	Use extra "buffer" week allocated at end of project to finish these requirements.	Will	Weekly

23	Project	Previous report/documentation poorly developed	Lots of changes will be required to get reports up to a base standard (from which extra content can be added).	3	3	9	During team meetings each week, team members can highlight aspects of the report that are underdeveloped. Then other team members can provide input and help during the meeting to help improve the section to an acceptable quality.	Ben	Weekly
24	Project - People	Team members get frustrated with legacy code and spend excessive time refactoring it.	This would lead to that team member becoming over-burdened with work and possibly missing deadlines (as they are too busy refactoring)	4	4	16	Other team members involved in the implementation can review the architecture and documentation of the inherited code base with the frustrated team member. This will help the team member understand the code so they can work on adding to it, instead of continuously refactoring it.	Will	Weekly
25	Project - Tools	Team members can't get tools previous team used working. (For example, tools like the LibGDX Texture Packer or the LibGDX Skin Composer.)	This would cause development to halt as the team spent time trying to install and use the tools instead of developing the project. If the tools are critical to the updating specific parts of the inherited project, it could mean key aspects of the system couldn't be updated/modified.	5	3	15	Contact university IT support for help in downloading the software, or to get software installed on university computers.	Will	Weekly (until the team has set up all tools previous team used successfully)
26	Project - Technology	Legacy code is not easily testable.	This could cause a low test-coverage in the project (meaning bugs could more easily slip in to the code). It could also mean that development of tests takes much longer.	4	3	12	Testing and implementation teams work together to come up with a modified architecture which can be more effectively tested.	Jamie	Weekly (as new areas of the code base have tests added to them)
27	Product - People	Team doesn't have skills required to replicate the graphical style of the original product	The final product will look disjointed and lack a cohesive presence. This could cause the product to be far less attractive to potential customers.	2	4	8	As there are a relatively small number of assets in the game, a new sprite pack could be found to give the game a unified aesthetic.	Seyi	Weekly (while new graphics are still being made for the game)

References

[1] I. Sommerville.(2015, Aug. 20). Software engineering [Online]. Available: https://eu.alma.exlibrisgroup.com/leganto/readinglist/citation/52275872540001381

[2] Office of the Chief Risk Officer (Stanford University). Definition of Risk Owner [Online]. Available: https://ocro.stanford.edu/enterprise-risk-management-erm/key-definitions/definition-risk-owner#:~:text=Risk%20Owner%3A%20The%20individual%20who,his%2Fher%20risk%20management%20efforts.

[3] K. Eby. (2018, Sept. 20). Agile Risk Register Template for Information Technology [Online]. Available: https://www.smartsheet.com/risk-register-templates