UNIVERSITY OF YORK

DEPARTMENT OF COMPUTER SCIENCE

Risk Assessment and Mitigation

Group 20

Formerly Group 16

Group Members:

Group 16

Charlotte MacDonald
Hollie Shackley
Luis Benito
Kaustav Das
Sam Hartley
Owen Gilmore

Group 20

Leuay Dahhane
Max Irvine
Sam Butler
Flynn Gadsden
Jacob Wharton
Billy Moore

a)
The risk management process is split up into four main sections: risk identification; risk analysis; risk planning and risk monitoring.

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.

Since we are taking on a brownfield project, our risks will emphasise the importance of quality with regards to transitioning our legacy code.Transition risks, implementation risks and legacy code risks will be heavily focused on. Our very first step in creating the updated risk register was to brainstorm where we wanted to take the code - in line with our requirements . Once discussed we identified initial risks and conversated with team 16 to confirm and answer any questions regarding the legacy code.This process clarified the specific risks we would face and subsequently we added them to the risk register.

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.

After identifying risks associated with our brownfield project, we focused on the following key areas:

- Transition Efficiency: Likelihood that we will be able to implement adequate knowledge transfer , for example ensuring all team members understand critical legacy information.

- Code quality: Likelihood that we will be able to implement high quality code , however this takes into account encountering challenging code that is poorly written or undocumented

- Compatibility issues:Likelihood of encountering issues regarding integration with the existing development systems and our new implementation systems, ensuring we don't encounter any conflicts.

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: regular code reviews and refactoring, which aim to identify and understand key aspects of the legacy code and plan our steps to implement new working code ; 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.

b) 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	(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.	Project manager	Biweekly
2	Project	(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.	Product owner	Weekly
3	Project	(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	Team leader	Weekly

4	Project	(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.	Project manager	Weekly
5	Project	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	Team leader	Weekly
6	Project	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.	Project manager	As required (during the time of absence)
7	Project	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)	Project manager and Team leader	Weekly, until return of absent member is confirmed
8	Project	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)	Project manager and Team leader	When and if it happens.
9	Project	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)	Project manager and Team leader	When and if it happens.

10	Project	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.	Project manager and Team leader	Weekly
11	Project	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	Project manager	Weekly (until its no longer a problem)
12	Product and project	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.	Product owner	As needed, based on frequency of changes in requirement
13	Product and project	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 (lacklustre product)	5	3	15	Negotiate deadline adjustments and time extensions. Also assess the resource reallocation and workload.	Project manager	Weekly (until it's no longer a problem)
14	Product	Inadequate testing leads to issues with the product	Lacklustre 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.	Product manager	Monthly
15	Product	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.	Product manager	Biweekly
16	Product	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.	Product manager	Weekly
17	Product	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.	Product manager	Weekly

18	Project	The development team can't grasp the previous code- may be due to bad communication from the previous team-	This can lead to poor integration of new features with the existing features or vice versa.	5	3	15	Ensure strong version control is carried throughout the development process .Enabling us to backtrack and fix errors if any arise	Project Manager Development team	weekly(until development is complete)
19	Project compatibility	Tools used in the greenfield project can't be used or aren't used effectively(can be tools like tiled maps)	If our implementation team can't use these tools effectively it can lead to issues developing/slow down the development process leading to poor code quality/time management.	5	1	5	Ensure all implementation members can run and work with the tools, if any issues arise contact university IT resources early so the implementation isn't slowed down drastically	Project Manager Development team	First week(assuming no issues arise , if any issues arise carry out mitigation)
20	Projec-t documentation	Documentation provided by the greenfield team is poorly written and requires a drastic change	This can be time consuming as the change report will have to be reworked. This can also lead to misunderstanding of the legacy code, for example a misrepresentation of the architecture. Further leading to more time being needed by the development team to understand the implementation	4	3	12	If there are any discrepancies with the change document and the implementation discuss it with the greenfield developers. Identify and estimate time taken to change the report if there are any issues so the team doesn't fall behind the time schedule.	Project manager and each	First week
21	project	existing bugs or issues in the legacy system that were not discovered or resolved	If bugs aren't identified before brownfield implementation begins bugs can arise later on in the development process and affect the stability of the code	5	4	20	We need to perform code analysis and ensure adequate unit tests are deployed, if this isn't the case already we must create these unit tests to ensure the components function correctly.	Project manager and development team	First week for the legacy code and weekly continuous testing for any new implementations
22	Project and product	our customer's requirements can't be implemented in our system	If we can't meet the expectation of our stakeholders our product will not be up to their standards , leaving us with an unsatisfied customer	5	3	15	In order to adhere to our customer we must involve them in our development process from an early stage, this allows us to solidify requirements and stay focused on producing a strong product and create a roadmap based of this	Project manager	In our first week consult the customer, design a roadmap and present it to them. Once they confirm the roadmap stick to it

23	Project and product	Ensure we don't stray to far from our legacy code	Our brownfield project relies on building on the legacy code , if we stray too far we won't be working within the constraints of our stakeholder	4	3	12	Ensure any drastic changes are run through with our customer , and once the roadmap(risk22) was implemented we stick to it , if any changes are necessary bring these to the customer before carrying out the implementations	Project manager Project dev team	As needed - depending on changes to roadmap
24	Project and product	Resources taken to implement all requirements is miscalculated	If we miss estimates we will fall behind on time , leading us to rush development. Sacrificing code quality and will therefore lead to creating a bad product	4	4	16	Our team will implement gantt charts (presented in team planning-plan.2).These gantt charts will be followed and updated accordingly,Weekly checkups will be carried out with the entire team to ensure help is given and no aspects fall behind .	Project manager	Weekly
25	Product	Our graphical representations aren’t in line with the legacy code	This will lack cohesive presence and drastically lower our code’s quality as textures and sprites aren't consistent	5	3	15	If we encounter this issue, before we change sprite packs or graphical representations we must inform our customer to ensure we don't stray too far away from the project to ensure we produce the required item	Project manager development team	Whenever necessary
26	Project -dependencies/ scalability	Dependencies in the legacy code may not be updatable , or break once new implementations are introduced	This will cause integration issues and lead to incompatible new implementations	5	5	15	We must first verify if all our existing dependencies are compatible with any new proposed implementations.As development continues we should keep backup files and implement continuous testing	Project manager Development team	First week for verification and then weekly for continuous testing.

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