Team iPatch

Assessment 1: Risk Assessment and Mitigation

Christian Pardillo Laursen, Filip Makosza, Joseph Leigh, Josh Wakefield, Mingxuan Weng & Oliver Relph

Introduction

Risk management is important because unexpected circumstances can happen which may hinder development or even cause the project to outright fail. Therefore, it is essential for us to identify the potential risks that affect our project and both reduce their likelihood and mitigate their impact. Taking account of the risks improves the project management and significantly reduces software rework which will reduce a large amount of cost.[1] In the table below, we are going to identify our risks, analyse them and plan around them. A member of the team will take the role of risk manager, whose job it is to re-assess the risks as software is developed and take action when unexpected consequences occur. This allows for other team members to better focus on their tasks while having a dedicated person on risk management will make them more observant towards these unexpected consequences.

To begin with, the risks will be identified and then classified as either project risks, product risks or business risks. These three type of risk will impact different parts of the project so categorising them will help software developers deal with the risk more easily. The guidelines for the different kinds of risks are as follows:

project risks: the risks which may affects the project schedule
product risks: the risks which may the quality or performance of the product (e.g.technical issues)
business risks: the risks that affect the client

After identifying the risks, we need to analyze them and describe their likelihood and the severity of their impact. Both likelihood and severity will be rated on a Low-Medium-High scale and have one of three highlighting colours (green, yellow and red).

Likelihood Low-Medium-High scale:

Low: has a low chance of happening if no action is taken
Medium: might happen if no action is taken
High: is likely to happen if no action is taken

Severity Low-Medium-High scale:

Low: estimated to cause a small amount of damage which is easy to overcome.
Medium: estimated to cause some damage which can be overcome and may cause some delay of several days.
High: estimated to cause serious damage to the project which is hard to overcome and may cause the delay of several weeks.

After analysing the impact level of each risk, risk planning is carried out. To minimise losses caused by the risk, a rough plan on how to mitigate and overcome the risk will be necessary. In the column of mitigation, we will describe how we will avoid the risk and how we will deal with it when it occur during the project.

The last step to risk management is risk monitoring. This is carried out by the risk manager throughout development, where the probability and impact of the risk are regularly

assessed, and in the case of a risk becoming a reality the mitigation plan is carried out.
During the implementation of our project, if a new risk were to emerge or any current risk were to be deemed no longer a risk then we would update the risk register appropriately, either by highlighting a risk as no longer relevant, or by adding a new risk to the table.

Risk Table

RISK ID: **1**	LIKELIHOOD: **Medium**	SEVERITY: **Low**
CATEGORY	Project	
DESCRIPTION	During the project a member in the group may fall ill unexpectedly and be unable to work for some time	
MITIGATION	The project plan should allow for delays which would cover the case where the absence is short. If the time is long, the tasks might need rescheduling to ensure they are completed on time	

RISK ID: **2**	LIKELIHOOD: **Medium**	SEVERITY: **Medium**
CATEGORY	Product	
DESCRIPTION	After doing a lot of work, the customer may suddenly change their requirements which will seriously affect the initial schedule of the development and may cause a delay.	
MITIGATION	Communicate with the customers regularly, and implement the changes in requirements as soon as possible. The project planning should allow flexibility to include the extra work required as seamlessly as possible	

RISK ID: **3**	LIKELIHOOD: **Low**	SEVERITY: **Medium**
CATEGORY	Project	
DESCRIPTION	Documents and resources are lost because of a technical issue.	
MITIGATION	Always keep backups of our project, never store critical code or documents only locally and push code to the remote server regularly.	

RISK ID: **4**	LIKELIHOOD: **Low**	SEVERITY: **High**
CATEGORY	Project	
DESCRIPTION	Lose one of the group members	
MITIGATION	Reassign workloads fairly such that no one group member is forced to take on the full brunt of the impact and the project can still be finished on time	

RISK ID: **5**	LIKELIHOOD: **Low**	SEVERITY: **High**
CATEGORY	Project	
DESCRIPTION	A requirement cannot be implemented because of an early decision regarding architecture	

MITIGATION	Construct the architecture around the requirements, be aware of what is blocked out when decisions are made.

RISK ID: **6**	LIKELIHOOD: **Medium**	SEVERITY: **High**
CATEGORY:	Business	
DESCRIPTION:	Misunderstanding of the requirements resulting in a product that doesn't fit the needs of the clients.	
MITIGATION:	Hold meetings often to ascertain that the requirements are understood correctly by both parties and good progress is being made towards them.	

RISK ID: **7**	LIKELIHOOD: **Medium**	SEVERITY: **Medium**
CATEGORY:	Product	
DESCRIPTION:	Team members lack the technical skill necessary to complete a task they have been assigned.	
MITIGATION:	Distribute tasks such that they are matched to the skills of the people performing them, account for learning curve when planning out allotted times for tasks.	

RISK ID: **8**	LIKELIHOOD: **Medium**	SEVERITY: **Low**
CATEGORY:	Project	
DESCRIPTION:	Decisions taken in group meetings are lost due to them not being recorded.	
MITIGATION:	Take notes of every important point in meetings and save them with a sensible naming system.	

RISK ID: **9**	LIKELIHOOD: **High**	SEVERITY: High
CATEGORY:	Project	
DESCRIPTION:	The estimates of the time required for tasks to be completed are not accurate and we can not finish the task on time or delay.	
MITIGATION:	Plan around the possibility of delays by allocating more time to each task than is thought strictly necessary.	

RISK ID: **10**	LIKELIHOOD: **High**	SEVERITY: **Medium**
CATEGORY:	Product	
DESCRIPTION:	There are still bugs in the final product.	
MITIGATION:	Implement rigorous product testing and reviews at every	

	stage of development to ensure no critical bugs are present in the finished product. Every artefact produced shall undergo a review before being added to the main program.

RISK ID: **11**	LIKELIHOOD: **Medium**	SEVERITY: **Medium**
CATEGORY:	Business	
DESCRIPTION:	Students, the target audience, do not find the game fun.	
MITIGATION:	Playtest with students at regular intervals during development to ensure the game is fun.	

RISK ID: **12**	LIKELIHOOD: **Medium**	SEVERITY: **Medium**
CATEGORY:	Project	
DESCRIPTION:	Focus is moved to implementing unnecessary features, neglecting the core game	
MITIGATION:	Keep track of the progress being made towards the core game being finished. Only implement extra features if development is ahead of schedule for the core game.	

RISK ID: **13**	LIKELIHOOD: **Low**	SEVERITY: **High**
CATEGORY:	Project	
DESCRIPTION:	Team members have different ideas on how the game should operate, leading to conflicts and problems during development.	
MITIGATION:	Establish main game elements before development and discuss any points that have not been brought up before before developing any artefact.	

RISK ID: **14**	LIKELIHOOD: **Low**	SEVERITY: **High**
CATEGORY:	Project	
DESCRIPTION:	The project tools we plan to use are not available.	
MITIGATION:	Carefully choose every tool to ensure it will be available and supported throughout development.	

RISK ID: **15**	LIKELIHOOD: **High**	SEVERITY: **Medium**
CATEGORY	Project	
DESCRIPTION	New Project has a different framework and so the developer team's members need to adjust/ learn how the new framework acts	
MITIGATION	The developer team should research the basics of other frameworks used and how they act differently to the framework we currently use in order to mitigate time wasted. Also we will assign a timeframe for the developer team to get accustomed to the new framework.	

RISK ID: **16**	LIKELIHOOD: **Medium**	SEVERITY: **Medium**
CATEGORY	Project	
DESCRIPTION	The new project contains bugs which are critical to the program	
MITIGATION	Assign a timeframe in the plan to allow for the developer team to familiarise themselves with the new project and fix any bugs which they can find fixes for in that time.	

RISK ID: **17**	LIKELIHOOD: **Medium**	SEVERITY: **Medium**
CATEGORY	Project	
DESCRIPTION	New project is undocumented or is very 'cluttered'	
MITIGATION	During our selection of the new project, choose a project with either sufficient documentation and modulation, or choose one which we feel we have a sufficient understanding of.	

As a team we feel that the risks we have outlined are very broad to our roles and so all team members have equal responsibility in ensuring their mitigation. Therefore the risk ownership for the risks outlined falls on every member .

References

[1] Boehm, B., Software risk management. In *European Software Engineering Conference* (pp.1-19).1989,September[Online]. Available:https://link.springer.com/chapter/10.1007/3-540-51635-2_29 [Accessed 23th November,2018]