SEPR

Assessment 1

Risk Management

JORDAN CHARLES
SAMUEL HUTCHINGS
CHLOE HODGSON
GOLNAR KAVIANI
TAMOUR ATLAF
JACK THOO-TINSLEY

2019/2020

Risk Assessment & Mitigation

Every software engineering project comes with risks which can damage the project. A risk is a future activity or event that may threaten the success of a software engineering project. However, a team can minimise the damage from risks with a robust Risk Management system. For our Risk Management to work well, we will largely be using qualitative analysis. The qualitative analysis can be broken down into the following sections: the identification, categorisation, a description, the likelihood of it occurring, its impact severity and any mitigation or contingency plans. [1]

To maximise the effectiveness of our Risk Management, each risk in our project has been identified by researching online and brainstorming in a group the threats to our project. We looked at websites for common threats to a software engineering project and discussed as a group what some specific threats could be to our project. Each risk was assigned a unique ID, type, likelihood rating, impact severity rating and an owner who is responsible for handling that risk.[2] Our General Risk Manager will be responsible for updating any additional risks discovered during our project. Risks that were clearly insignificant will not be included. Each risk is given an ID based on what risk type it is categorised under. If it can be categorised under multiple types, the type that we believe is most appropriate will take priority.

Risk type, the types of risk and impact of that risk is defined as:

- **Project** *(PJ)* - Risks that affect the project schedule and project resources.
- **Product** *(PD)* - Risks that affect both the functional and non-functional requirements thus the quality of the developed product.
- **Business** *(B)* - Risks that affect the organisation of developing and procuring the software.
- **Technology** *(TN)* - Risks relating to the hardware or software used in the project.
- **People** *(PP)* - Risks relating to team members within the project.
- **Tools** *(T)* - Risks relating to the software and tools used in the project.
- **Requirements** *(R)* - Risks relating to changes in customer requirements while the project is ongoing.
- **Estimation** *(E)* - Risks relating to the estimation of the system characteristics and system resources

The likelihood and severity rating, the probability of a risk occurring and the impact of the risk if it were to occur, will both have three levels:

- Low - Unlikely to occur/Little to no impact.
- Moderate - A fair chance of occuring/A fair amount of impact.
- High - An extremely high chance of occurring/A large impact.

We decided to distribute the risk ownership between three team members to ensure each risk could be constantly monitored. Risks that can be detrimental to the project are assigned more than one manager. As we chose Agile Development Method, the risk monitoring will be done every week in a meeting, each risk manager will discuss any concerns regarding risks and any problems caused.

The following roles have been assigned to distribute risk ownership:

- Project Manager (PJM) - Chloe Hodgson
- Product Manager (PDM) - Jack Thoo-Tinsley
- General Risk Manager (GRM) - Tamour Altaf

As Product and Project were common risks, they have their own managers to tackle those risks and the General Risk Manager will assist in tackling more severe and likely risks.

Risk Planning

ID	Risk Type	Description	Likelihood	Severity	Mitigation/ Contingency	Owner
PJ1	PJE	Misestimation on development and scheduling times.	Low	Moderate	Consistent use of Jira and its sprint system will be used to stay on top of deadlines.	PJM
PJ2	PJPP	Members of the team are unavailable to contribute to the project for a prolonged period of time.	Low	High	Avoidance of a low “bus factor”. Each task will be shared between more than one person such that there is no complete dependency on one person.	PJM GRM
PJ3	PJPP	A member of the team is being uncooperative or not contributing their fair share of the workload.	Low	High	Bi-weekly meetings to check on workload, progress and wellbeing of all team members.	PJM GRM
PJ4	PJPP	Disagreements between team members	Low	High	Constant communication to make sure all team members are happy. Resolve conflict as soon as possible or contact module supervisor if conflict is serious.	PJM GRM
PJ5	PJPP	The workload on an individual team member is too great resulting in a decrease in productivity.	Low	Moderate	Bi-weekly checkups. If a team member is struggling with the workload it can be redistributed accordingly.	PJM
PJ6	PJPP	Overcomplication of simple tasks.	Moderate	Low	Constant peer review on the latest project version.	PJM

PJ7	PJ PP TN T	Learning process of new software or programming languages takes more time than anticipated	Low	Moderate	Our project manager will be responsible for introducing team members to any new software used. Meetings will be called if team members are still having trouble learning any new software or programming language.	PJM
PJ8	PJ TN T	Files are lost, misplaced or deleted.	Low	High	Backing up documents on different computer systems or online software. Making use of GitHub as it's reliable and having version control throughout the project.	GRM PJM
PJ9	PJ R	A change in requirements midway through the project.	Moderate	Moderate	Clear and concise documentation with all iterations available to everyone through Google Docs and Github. This allows easy changes.	PJM
PJ11	PJ PP	A team member is lacking in motivation or is unsure of their current task.	Moderate	Moderate	If a team member is not enjoying their current task and it is affecting the development schedule then tasks can be re-assigned to better accommodate this.	PJM
PJ12	PJ PP	Other priorities such as university work and social activities may reduce the amount of time going into the project.	High	Moderate	Setting aside bi-weekly meetings and individually an allotment of time ensures that work is done on the project.	GRM PJM
PD1	PD PP	Final product does not meet the	Low	High	Regular checkups and peer reviews on work done always keeping in	PDM PJM

		requirements and specifications.			mind the requirements.	
PD2	PD TN T	Relevant technology goes down resulting in the loss or inaccess of our work. E.g. Jira	Low	High	Where possible any documentation is backed up on more than one software.	GRM PDM
PD3	PD T PP	Risk assessment fails to identify all relevant risks.	Low	Moderate	Within the risk managers, review any new risks that have a slight chance of occurence are noted and added to the assessment.	PDM
PD4	PD PJ PP	Inconsistent documentation	Moderate	Moderate	Documentation will be reviewed weekly by the team to avoid any inconsistencies.	PDM
B1	B PD	Final product contains code bugs.	Low	High	Group testing every week to solve small and large bugs. Regular team meetings and monitoring of the product. Contact module leader if a bug cannot be solved by the team.	PDM GRM
B2	B R PD	Shareholders do not like the final product.	Low	High	Regular communication and always checking that the direction our project is heading is in line with the vision of the shareholders.	PDM GRM

References

[1] P. Simon, D. Hillson and K. Newland, Project Risk Analysis and Management guide. Norwich: APM Group Limited, 1997.

[2] "Risk Management in Software Development and Software Engineering Projects", Castsoftware.com, 2019. [Online]. Available: https://www.castsoftware.com/research-labs/risk-management-in-software-development-and-software-engineering-projects. [Accessed: 01- Nov- 2019].