1.0 Introduction and Justification

"Risk management is a series of steps whose objectives are to identify, address and eliminate software risk items before they become either threats to successful software operation or a major source of expensive rework" [1].

In order to identify risks to our project we followed a risk management cycle of identification, analysis, prioritisation, planning & mitigation and monitoring. As a team we decided to adopt the condition-transition-consequence (CTC) format [2] to allow a standardised systematic approach to defining risks as they arise. Initially, in order to identify risks we engaged in a team discussion and brainstormed as many risks as we could.

Next we categorised the risks into project risks and product risks by the definitions given in Laurie Williams' Risk Management [3]. Product risks will be identified and monitored as the project is implemented.

Then we assessed the probability and impact of a loss occurring by modifying and utilizing a standardised scale to reflect the perceived likelihood of the risk and impact of the loss if it were to occur. As the original scale was too complex we simplified it to better suit the scope of our project. Each team member assessed each risk, then fed back their assessments to the group and debated their thought process until a general consensus was reached. This process was modelled on the Delphi Technique [4] and aims to reduce the subjectivity of classifying a risk.

The team meets regularly to review progress and risks. In order to monitor the risks each team member is able to communicate their concerns back to the Risk Manager and the team via Slack or in any of the meetings during the week. We set out with the intention for each team member to keep their own list of the 10 risks they thought to be *most likely for their roles?* as well as making a top 10 list of risks to the project in general. Each team member was asked to update their list weekly to reflect how they perceived each risk's impact and probability of occurring as time went on. Based on these individual lists, the plan was then to update the overall team list on a weekly basis to reflect the changes the team identified. The risks with a higher priority are shown at the top of the list. We manage probability in a similar way, if the probability is rare it is deemed not important enough to be actively managed, and therefore is monitored. This better fits our agile approach to rest of the project; majority of tasks are completed by multiple team members as opposed to individual work and as such keeping track of team risks became the more intuitive option.

A Risk owner for each risk was selected from the team, the individual was chosen so that their experience would provide insight into the risk they are managing. This aims to have one individual carry out the majority of risk management for an individual risk. This was decided as we found that too many people attempting to manage the same risk was counterproductive, and and also became a risk.

Next the team decided on mitigation techniques for risk avoidance and reduction feeding back in a similar iterative style to categorisation. This approach to risk mitigation fits with our agile methodology as tasks are mainly completed in groups, and as such we have identified methods for mitigating and avoiding risks working as a team. For example we use Google Drive which backs up all of our work in the cloud and as such reduces the risk of work being lost by damage to physical devices such as USB sticks. Each team member was educated about the different types of risk mitigation (acceptance, avoidance, control, transfer and monitoring) in order to develop their ability to mitigate risks as they arise without the help of the Risk Manager. We used a classification to give a rank to each risk, which is used to manage the mitigation process.

We decided that the risk log will be maintained in tabular and colour coded format. To do this we have created a hybrid format that better suited our needs and is based on the risk management matrix [5] and the risk table template from Laurie Williams' Risk Management [3]. The colour coding in the risk management matrix gives a clear visual representation of risks that can be monitored and makes distinguishing between different risk categories easier and faster; whilst the risk table template from Risk Management [3] lets us rank the risks and provide a mitigation strategy. Our hybrid format hence maintains benefits from both methods. This should streamline the whole risk management process during the project by making updating the risks document easier.

Bibliography

[1] B. Boehm, "Software Risk Management," IEEE Computer Society Press, 1989.
[2] D. P. Glutch, "A Construct for Describing Software Development Risks," July 1994.
[3] L. Williams, “Risk Management”, 2004
[4] U. Gupta and R. Clarke, "Theory and Applications of the Delphi Technique: A bibliography", Technological Forecasting and Social Change, pp. 185-211, 1996.
[5] SEPR, "Rank Table," 4 November 2016. [Online]. Available: https://seprated.github.io/Assessment1/RiskTable.jpg [Accessed 21 January 2017].