**Risk Assessment and Mitigation**

**Stakeholders:** Richard Paige, University of York Communications Office

**Team:** Barney Morgan, Cameron Smith, Harry Berge, Jake Phillips, Matthew Wilkie, Rob Weddell

**Method**

The process of eliciting risks began with a team meeting in which all possible risks were researched [1, 2], discussed and noted down. This list was then refined to ensure little overlap and to eradicate the unnecessary risks (either extremely unlikely to happen or risks so mild that they would cause no noticeable impact). After multiple iterations of merging similar risks together and removing extra risks, a concise risk table was produced.

Following this, each risk was assigned to a specific team member whose expertise and allocated role most suited the risk and its mitigation. The job of this team member was not necessarily preventative action but rather to mitigate the issue according to our assessment if the risk were to occur. This ensured that when a problem did arise, there was a dedicated member of the team focused on fixing it as soon as possible without any fuss. (Key: PM - Project manager, S - Secretary, CI - Client interface, SA - Software architect, WD - Web developer, TC - Technical Consultant)

The values of each specific risks impacts and the likelihood of that risk occurring were then created from an average of each team member’s personally assigned values. Following this, a meeting was held to discuss the results of the risk assessment and to potentially correct any outliers or errors in the values. Should a value need changing, a group consensus was required before any alteration. After further research [3], it was decided to stick to a relatively simple method of ranking the likelihood and impact of each risk given the relatively small nature of the project and the fact that the software being developed is strictly non-critical. Thus, a simple 3 tier system was proposed to rate the likelihood, and a 5 tiered scale was given to the impacts. The team felt it was unnecessary to go into any more detail (for example the PRINCE2 risk register [4]) as there was no need to split columns further than had already been done as one general risk rating would suffice for the project. By keeping it simple like this, it enabled the team to apprehend the risks with ease at a glance and thus made the list more effective at both mitigating and preventing risks. Risks were ranked as such:

* Likelihood - (Green - Red)
* Green - Fairly unlikely to occur
* Yellow - Mildly likely to occur
* Red - Most likely will occur
* Impact - (1-5, one being the lowest)
* 1 - Not very impactful (Most likely will not even be noticed as a setback)
* 2 - Very small impact (Will be noticed but will have little effect on the assignment)
* 3 - Mild impact/setback (Will take people time to fix but won’t be seen as too much of a problem)
* 4 - Quite significant impact/setback (Will cause a lot of disruption to the assignment)
* 5 - Massive impact/setback (Possibly assignment breaking level of damage)

Several categories were created in order to clearly attribute risks with the team’s assessment of them. Risks can be assigned any number of categories. The categories and definitions are as follows:

* **TOOL** - This risk is mitigated by our choice of tool for this aspect of the project.
* **COMM** - This risk is related to communication between group members.
* **PROD** - This risk illustrates a potential degradation in quality to the end product
* **SCHED** - This risk may mean a delay in our project schedule. Internal deadlines may need change.

In the event of a new risk being discovered, a similar process will be repeated by which we hold a team meeting where we discuss the cause of the risk and the best way of mitigating the newly discovered risk. Once the risk mitigation has been discussed, the likeliness and impact can then be discussed as well as the owner of the risk. This information should all be documented in the risk table to provide the team with a table with as much useful information as possible.

**References**

[1] ITProPortal, Top 10 Software Development Risks [Online] Available: <https://www.itproportal.com/2010/06/14/top-ten-software-development-risks/> [Accessed 30 Oct. 2018]

[2] Existek, Top 5 Risks in Software Development [Online] Available: <https://existek.com/blog/risks-software-development/> [Accessed 30 Oct. 2018]

[3] V. Antinyan et al, Defining Technical Risks in Software Development [Online] Available: <http://web.student.chalmers.se/~vard/files/Defining%20technical%20risks.pdf> [Accessed 30 Oct. 2018]

[4] PRINCE2 Risk Register Wiki [Online]. Available: <http://prince2.wiki/Risk> [Accessed 1 Nov. 2018]