Risk Management Process and Justification

Group 9

Pluto Pioneers

James Lawson, James Rooke, Dema Williams, Nicholas Barker, Sam Slade, Joshku Gauntlett, Seiya Ikeda

The risk management process our group has followed is a 4 step process of identifying risks, performing a risk analysis by subjectively deciding the likelihood and severity of the risks, planning for those identified risks by suggesting ways to mitigate and/or prevent those risks, and monitoring the risks by assigning members to help plan for the risks [1]. We have chosen this system as we feel it best helps with mitigating and/or preventing possible risks we may face and makes sure those risks are dealt with. We randomly and equally distributed the risks to each person, making them the owner of the risk. This makes sure no one would be overwhelmed with managing risks leaving them unable to do other tasks.

Our risk register displays a unique ID for each risk which enables us to quickly reference that it, a short description of each risk, type, severity and likelihood, avoidance and/or mitigation (certain ones may not be avoidable so they will only have a mitigation), and the member that will monitor the risk. The risks are grouped by type (Estimation, organisation, people, etc…). This format displays all the necessary information for each risk and in a clear way.

Our risk register is below:

Risk ID	Description	Risk Type	Severity	Likelihood	Avoidance	Mitigation	Owner
1	The time required for a stage in the software engineering process is underestimated	Estimation	Moderate	Moderate	Give a bit more time for each stage than we think we'll need.	Consider a reduction in the project scope, whilst still trying to deliver for the customer	Dema Williams
2	The university IT sytems go down.	Organisational	Low	Low		Work from home	Nicholas Bar…
3	Team members become unavailable	People	Low	High		Ensure there more than one person is familiar with each aspect of the project	Sam Slade
4	Team members do not have the required skills		High	Moderate	Make sure there is at least one person that has the skills is contactable.	Allow for time to learn new skills	James Rooke
5	Team members do not get along		Low	Low	Team building. Regular check ups.	Team building	Josh Gauntlett
6	The requirements change during the development of the software	Requirements	High	High		Prepare for this by thinking of requirements that may be added or changed and how we may implement them. Follow agile principles	Seiya Ikeda
7	The chosen game engine is difficult to learn	Technology	Moderate	Moderate	Carefully select a suitable game engine	Allow for extra time to learn how to use the game engine	James Lawson
8	The chosen game engine does not support a required feature		High	Low	Carefully select a suitable game engine	Perhaps find a work around / third party library	Dema Williams

9	The chosen IDE is difficult to work with	Tools	Moderate	Low	Carefully select a suitable IDE	Allow time for learning how to use the IDE	Nicholas Bar…
10	Persons work gets corrupted or lost		High	Low	Ensure work is uploaded to group repositories	Make backups of work	Sam Slade

Reference list

[1] *Sommerville, I 2015, Software Engineering, Global Edition, Pearson Education, Limited, Harlow. Available from: ProQuest Ebook Central*