Risk Assessment & Mitigation Risk Rating

To determine the overall damage to the project that a risk may have, we applied the likelihood and impact of the risks via a risk matrix to abstract a ‘risk rating’, ranking them from Low, Medium, High and Critical. These risk ratings are selected using the 5x5 matrix below that divides impact and likelihood into 5 levels. For our project a 5-level rating scheme is more suitable than a standard 3-level rating
(low-medium-high) because it allows a more refined assessment of risks. A higher level ranking, e.g. a 9-level rating, could’ve been chosen but given the nature of the project, a game - not a critical system, this would have been too detailed. The impact and likelihood definitions are listed below. The number associated with impact and likelihood is summed together to determine the overall rating for a particular risk; 2-4 is low, 5-6 is medium, 7-8 is high, 9-10 is critical.

Likelihood	**Impact**				
	Negligible(1)	Minor(2)	Moderate(3)	Major(4)	Severe(5)
Very Likely(5)	Medium	High	High	Critical	Critical
Likely(4)	Medium	Medium	High	High	Critical
Somewhat Likely(3)	Low	Medium	Medium	High	High
Unlikely(2)	Low	Low	Medium	Medium	High
Very Unlikely(1)	Low	Low	Low	Medium	Medium

Impact Definitions
- Negligible - The impact of the risk will only delay the project by a small number of hours
- Minor - The impact of the risk will delay the project by 1-2 days
- Moderate - The impact of the risk will delay the project by 3-5 days
- Major - The impact of the risk will delay the project by 1-2 weeks
- Severe - The impact of the risk will delay the project by 2 or more weeks

Likelihood definitions
- Very Likely - Occurs once a week
- Likely - Occurs once in two weeks
- Somewhat Likely - Occurs once in three weeks
- Unlikely - Occurs once a month
- Very Unlikely - May occur once during the project completion time

Risk format justification

The risks have been formatted into a table consisting of 5 columns; Risk Number, Risk Category, Risk name and description, Mitigation, and Overall Risk. The rows are sorted into descending order of overall risk rating so risks that would have the biggest impact are clearly visible. Risks are ranked by evaluating negative impact to project and likelihood of possibility. The table suggests mitigation strategies, that would help effectively solve potential problems. Risk Category shows which aspect of a project the risk is covering. The categories consist of; requirements, project complexity, planning and control, team, and organisational environment [1]. They show the spread of risks across all areas of the project and assists with assignment of risk ownership. Team leader owns risks related to organisational tasks of functioning of a group. Scrum master, that is assigned for a duration of each sprint, is responsible for handling all risks related to software development work. Some of the risks need to be handled individually by each team member.

Risks

Nr	Risk category	Risk	Mitigation	Overall Risk
1	Planning and control	Ineffective communic ation between group members	GitHub and Slack allows effective communicate between group members. Failure to communicate will have a reason. Schedule a special meeting with an elected mediator to allow poor communicators to explain their issue(s) and re-integrate with the team.	Unlikely/Severe
2	Require ments	Project team misunderst ood requirements	Following Scrum methodology frequent team meetings are held and requirements are often reviewed. Team maintains consistent communication via Hangouts and through face-to-face means.	Likely/Severe
3	Planning and control	Team members missing important team meetings	Communicating via Hangouts. Team members are provided with all necessary information that they missed. If that person is still missing team meetings, the problem will be reported to lecturers to be solved via official channels.	Likely/Major
4	Requirm ents	Inadequate architecture, performance and quality	Testing of the product should be carried out continually to ensure performance requirements are met.	Unlikely/Severe
5	Planning and control	Poor project planning	Our agile development model allow us to establish a clear overview of bottlenecks in the project and then refactor the plan as we go.	Unlikely/Maj or
6	Team	Deletion of project code	GitHub minimises the likelihood of complete loss of code, regular backup via forking to personal repositories increases redundancy and cloning of the repository to an offline location before major changes further reduces	Very Unlikely/Severe

			risk.	
7	Team	Certain team members lack of specialized skill required by the project	Tasks are divided up and given to the team member with the most experience in this area. This allows the tasks to be completed by somebody competent in the area.	Likely/Moderate
8	Planning and control	Poor productivity	Number of commits and number of additions/deletions in GitHub is used. It shows visual display of effort that people put in. It indicates team members that don't contribute enough. However, as some work is not done on github, it may not be that a low number of commits indicates low productivity.	Somewhat Likely/Moderate
9	Team	Member of team is sick and can't participate in further work	Program code is submitted in GitHub therefore the rest of a the team can continue working. No member will ever have a 100% share in a mission critical task. Frequent commits and clear comments will allow members to compensate for illness or absence.	Likely/Moderate
12	Planning and control	Project progress not monitored closely enough	Frequent team meeting are held to review progress. Gantt and burn-down charts are used to give a close view of project progress	Unlikely/Major
13	Planning and control	Project milestones not clearly defined	Milestones analysed within the planned scrum meetings, milestones can be re-arranged and definitions updated within a week of issues arising.	Somewhat Likely/ Moderate Impact
15	Require ments	The wrong software functions are developed	Customer requirements are analysed. User survey reviewed. Prototype tested. Discuss and resolve software issues in meetings.	Somewhat Likely/Moderate
16	Require ments	Continually changing requiremen ts	Changes are accepted as a fact of software projects. Prioritisation sessions are scheduled that allow changes to proceed.	Very Likely/Minor
17	Require ments	Incorrect system requirements	Task descriptions are reviewed frequently. Note that while our requirements are correct in the context of our current concept, that concept may change in the future.	Very LIkely/Minor

18	Planning and control	Gold plating (added features are not useful)	Requirements are frequently reviewed and tasks are worked on in order of priority. Prototype is created and tested.	Unlikely/Moder ate
19	Project complexi ty	Functionali ty is complex to implement	Team meeting is held to discuss the problem and find solution. Research is done on programming techniques or available software that would allow to solve the problem.	Somewhat Likely/Moderate
20	Project complexi ty	Problems to integrate separate pieces of code	The GitHub pull request system combined with a potential sit down meeting with all developers affected by the conflict in question will likely ensure a smooth development process.	Somewhat Likely/Moderate
21	Project complexi ty	Overriding each other's work	Over-writing other's work should only occur after proper code review enforced by the pull request system. This review process allows us to ensure any changes removing or changing others' work to be intended and functional.	Unlikely/Major
24	Team	Team member's local copies of project files are compromis ed after having been worked on heavily (prior to committal)	Submit commits immediately after each task is completed so that all completed work is backed up frequently. Use Google Drive to write up documentation and automatically save it on Google's servers.	Unlikely/Major
25	Planning and control	Created software code has a bug	Software code is saved on GitHub. It allows to access previous version of code and locate where the bug was introduced. A possible integration of Travis CI will allow immediate testing and debugging of code.	Very Likely/Minor
26	Planning and control	Something is not working as planned in sprint plan	Team is having regular sprint meetings, during which arisen issues will be discussed. Team members will split additional workload in order to resolve problem and to not fall behind with current plan. If it is necessary additional time is allocated and sprint plan is changed	Very Likely/Minor

27	Require ments	User interfaces do not fit needs	Prototype is created, scenarios are development. Customer description reviewed.	Likely/Negligibl e
28	Planning and control	Inadequate estimation of required resources	Frequent meetings are held. If needed additional resources can be allocated. Tasks can be divided between more group members if the velocity of one member is dropping due to tackling a large task alone.	Unlikely/Minor

References

[1] Tharwon Arnuphaptrairong "Top Ten Lists of Software Project Risks : Evidence from the Literature Survey" International MultiConference of Engineering and Computer Scientists 2011 Vol 1, IMECS 2011, March 16 - 18, 2011, Hong Kong. [Online]. Available: http://www.iaeng.org/publication/IMECS2011/IMECS2011_pp732-737.pdf