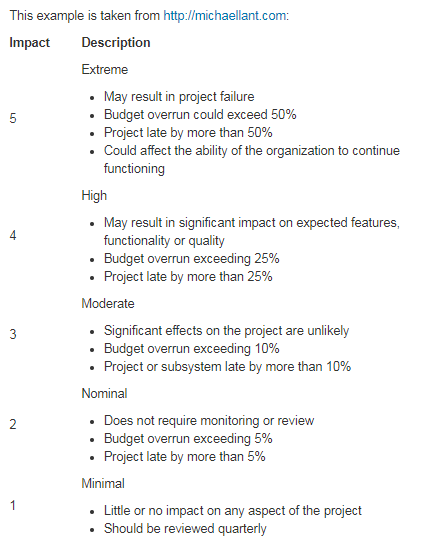
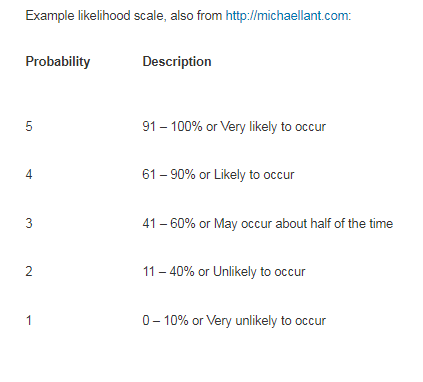
Risk Register





(taken from reading materials <https://moodle.vle.monash.edu/mod/book/view.php?id=5304126&chapterid=408149>)

**Organisational Risks**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk** | **Impact** | **Likelihood** | **Monitoring Strategy** | **Mitigation** | **Contingency** |
| Unrealistic targets from client | Not able to achieve the goals(4) | 3 | Making proof of concept | Client meeting and negotiation | Adjust goals to a reasonable range |
| Unrealistic targets from developers | Not able to achieve the goals(4) | 3 | Making proof of concept | Team meeting, estimate the possibility of implementing the program | Adjust goals to a reasonable range |
| Poor time management | Not able to deliver on time(4) | 2 | Set a countdown | Routine check on project process | Adding extra working time |
| Team members become ill | Making the project behind schedule(3) | 2 | Regular health check on each team member | Ensure the team is cross functional | Another team member picks up the work of the sick team member |
| Loss of team member | Lack of team member puts added workload on rest of team resulting in delays or unsatisfactory work (5) | 1 | Team communication on future plan | Team meeting and communication | Prepare substitutes |
| Not enough opportunity for client communications | Client does not have enough impact on the project leading to an unsatisfactory product (3) | 2 | Regular checking on product target, make sure the goals are similar | Regular program delivery | Increase client meeting time and frequency |
| Misunderstanding user requirements | Code is written that needs to be changed or replaced that leads to delays (4) | 2 | Regular program delivery | Regular program delivery, check on functionality | Client meeting and requirement review |
| Low motivation | Team fails to meet requirements on time (3) | 3 | Periodically check on everyone’s progress, note anything behind schedule | Regular check on schedule | Team activity and enough rest time provided |
| Changing requirements | Changing requirements result in code being useless or needlessly changed leading to delays (3) | 4 | Client contacts team and changes requirements | Build flexible program | Build more flexible program |
| Failure to manage client expectations | Team is unable to meet unrealistic goals and fails to meet requirements (4) | 3 | Client expresses disappointment with the team's progress | Client meeting | Make no agreement on unachievable targets, replace them with alternative features |
| Poor communication within team | Team members do not have access to all information and do not work together effectively leading to delays (3) | 2 | Team members not using chat or not showing up to meetings | Regular expectation check | Team meeting |

**Technical Risks**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk** | **Impact** | **Likelihood** | **Monitoring Strategy** | **Mitigation** | **Contingency** |
| Technological issues/failures | Unable to work on project at home or loss of data (2 to 5) | 2 | Someone alerts the group that their computer has ceased operation | Everyone maintains and checks their own hardware periodically | Use of university computers / labs  Backup work on GitLab and google drive |
| Insufficient knowledge | Work not completed on time or sufficiently (3 to 5) | 2 | Someone tells the group that they do not have the required knowledge | People watch lectures, learn from and teach their teammates | Ask for help and check university resources or internet |
| Insufficient hardware | Unable to do work at home (5) | 2 | Someone alerts the team that their computer is unable to perform required actions | Everyone checks and maintains their own hardware periodically.  Everyone supplies their own sufficient hardware if possible. | Use of university computers / labs |
| API changes, e.g. loss of support | Aspects of program become broken and have to be re-written (5) | 1 | Program stops working and group checks online to see api has changed | Regularly check API for future changes as used | See if previous versions are still available |