Risk Register

We Showed Up

**Legend**

**Impact**

5: Extreme

* May result in project failure
* Budget overrun could exceed 50%
* Project late by more than 50%
* Could affect the ability of the organization to continue functioning

4: High

* May result in significant impact on expected features, functionality or quality
* Budget overrun exceeding 25%
* Project late by more than 25%

3: Moderate

* Significant effects on the project are unlikely
* Budget overrun exceeding 10%
* Project or subsystem late by more than 10%

2: Nominal

* Does not require monitoring or review
* Budget overrun exceeding 5%
* Project late by more than 5%

1: Minimal

* Little or no impact on any aspect of the project
* Should be reviewed quarterly

**Probability**

5: 91 – 100% or Very likely to occur

4: 61 – 90% or Likely to occur

3: 41 – 60% or May occur about half of the time

2: 11 – 40% or Unlikely to occur

1: 0 – 10% or Very unlikely to occur

(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 | Team is unable to meet requirements | Client meeting and negotiation | Adjust goals to a reasonable range |
| Unrealistic targets from developers | Not able to achieve the goals(4) | 3 | Team is unable to meet requirements | Regular team meetings, estimate the possibility of implementing the program | Adjust goals to a reasonable range |
| Poor time management | Not able to deliver on time(4) | 2 | Goals aren’t reached on schedule | 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 | Other team members pick 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 member communicates they can no longer work on project | Prepare substitutes and backups for each task | Split the team members workload equally and appropriately between members |
| 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 | Schedule time for regular client meetings or let client choose how much communication they want with the team | 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 | Client is unsatisfied with the project | Regularly consulting with client on progress and decisions made on project | Client meeting and requirement review |
| Low motivation | Team fails to meet requirements on time (3) | 3 | Team member falls behind schedule or completes work at last minute | Periodically check on everyone’s progress, note anything behind schedule | Regularly check up on team member and help them schedule times to work on project |
| 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 | Discuss changing requirements with client and compromise to make them more feasible |
| 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 called to specifically address the issue of why communication is poor and fix them |

**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  Backup work on GitLab and google drive | Use of university computers / labs |
| 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 |
| API cannot be used in the way required | Some requirements cannot be met (5) | 5 | When the team is unable to implement the requirements with the API | Know the limits of the API and what can and can’t be done | Find other ways to implement requirements, learn to use the API in a different way to meet the requirements or negotiate change in requirements with client |