|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Response Strategy | Likelihood of Risk | Impact | Risk Level |
| Time management | Some work should be done regularly and updates to the project should be done more frequently so that it does not fall behind. Failure to adhere to this can result in a larger workload closer to the deadline which can cause errors and requirements not being met. | Moderate | High | Moderate |
| Estimation and scheduling | Planning the time frames for each sprint needs to be done efficiently so that development time isn’t lost. There needs to enough development time allocated so that when mistakes and errors do occur, there is enough time to fix them. This can be done effectively by monitoring existing projects and knowing how long each task can roughly take. Doing this enough will ensure more accurate estimations. | High | Small | Small |
| Procedural risks | Day to day work and conflicting priorities can hinder the project. To minimise this risk, some or little work at the least should be done each day without fail so that even when you do have a large workload some days, the project is still being worked on regardless. | High | Moderate | Moderate |
| Sudden growth in requirements | As the project progresses, issues that were not identified earlier can sometimes create problems towards the end of a project. To minimise this risk, when designing the project, it is better to think big and include every little detail. It is better to anticipate the worst-case or heaviest-use scenarios so that if something does become apparent later on, it shouldn’t pose too much of a risk. | Low | Moderate | High |
| Compromising on designs | Since designing is the most critical part of software development, ample time should be taken when going over this. Failure to adhere to a proper design process can lead to a lack of understanding of the requirements and in turn can result in catastrophic risks. | Low | Very high | Catastrophic |
| Gold plating | Adding extra features which is not specified by the brief can result in a waste of time during development. To minimise this risk, if extra features are to be added, the basic system and functionality should be developed first and if there happens to be time leftover afterwards, the extra features can be added. |  |  |  |