|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Risk | Description | Likelihood | Impact Level | Responsibility | Response | Control Measure |
| 1 | Poorly planned sprint and misjudged story points and MoSCoW prioritisation | 6 | 5 | SCRUM master | Follow agile methodology and adjust future sprints to accommodate changes | Follow agile methodology |
| 2 | Dev virtual machine goes down losing files | 2 | 10 | AWS | Consistently commit and push to GitHub | Version control(keeping up to date) |
| 3 | Dev pc crashes | 1 | 8 | Developer | Move on to a different machine as soon as possible | Version control – Consistently push to GitHub so minimal loss to project |
| 4 | Broken version of code pushed to dev branch on GitHub | 4 | 8 | Developer | Revert branch to most recent working version | Automate testing before for before pushing up |
| 5 | Push to main branch – If latest commit is broken will break the current working code | 2 | 10 | Project Owner | Revert branch to previous commit | Dev branch created to run parallel to main so if any mistakes are pushed up then it won’t break the current working code |
| 6 | Stats not up to date | 4 | 6 | Developer/user | Update to correct mistakes as soon as found | Review stats regularly |
| 7 | Players teams not up to date | 3 | 6 | Developer/user | Update db as soon as possible | Stay up to date on player trades |
| 8 | Instance goes down | 3 | 7 | AWS | Use temporary local db until new instance is up | Back up data |