

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Risk | Effect | Probability x/10 | Severity  x/10 | action | After action prob | After action effect |
| Planning too simple | Uncoordinated action | 3 | 4 | Review plan for coverage | 1 | More comprehensive plan |
| Database schema unsuited to task | Overly complex code to compensate | 4 | 7 | On code workarounds apperaing review relavent schema | 2 | More robust schema and simpler code |
| Bugs found on iser input | Application faliure | 6 | 7 | Code the test first. Only code for what has a test already | 1 | Good test coverage, bugs unlikely |
| Newness to git leads to complex/incorrect use of branches | No clear master branch to ship/review | 8 | 8 | Learn more git | 3 | Better knowledge of git leads to more robust version control |
| Sidetracked by "nice to haves" | Project wide but not deep. Does not fully function | 5 | 9 | Only action what is on knaban board todo. | 2 | Likelehood of shipping a viable product increased |
| Deleted objects may have a need to be recalled later | Would require complete rebuild of db object from external source | 2 | 3 | Include "delted flag" to omit  From results  Nb: would require a "delted items interfacec" .. | 2 | Acidental deletions have a fall back to restore from |
| User stories overly complex | Singular tasks have too many steps to comprehend simply | 7 | 4 | Limit user story checklists to 4 | 2 | Simple user storys lead to simple code |
| Varible names get to personal | Understanding code from new view is complicated | 6 | 1 | Define or use existing naming scheme | 1 | Another developer incoming would be able to better understand the code |
| Zero CI knowledfe | Project failiure | 10 | 10 | Pay atttention to ci learning | 2 | Able to use CI and also pass the project |
| User interface hard to understand | User confusioin and dissatifaction with deliverable | 4 | 4 | Attempt to conduct user testing on interface (ongoing) | 2 | Good communication with end user leads to more usable result |
| Use of password hardcoded or passed to github | Database hijack | 6 | 9 | Never include passwords in code!!! | 2 | More secure db |
| Artefact repository manager | Unknown | X | X | Learn about what this is | X | Unknown |
| Testing framework knowledge insufficent | Planned tests are not implented correctly / to best practice | 7 | 9 | Continue learning junit past my current limited knowleddge | 2 | Tests correctly cover code and correctly test code units |
| Local hardware failure | Unable to progress project past failure point | 2 | 10 | Be aware of temp replacement option from QA / ready to source own replacemetn | 3 | Time lost on falure can be made up . If time lost is not too great |
| GCP faliure | Stroed data is lost | 1 | 3 | Always keep backups of scripts used for DB creation, or additions | 1 | Although unlikely by proper data security this can be mitigated to near null |
| Maven issues | Project fails build | 6 | 9 | Only pass clean projects to git and commit regularly | 7 | Project issues take a long time to fix |