| **Risk ID** | **Risk Title** | **Risk Description** | **Likelihood** | **Impact** | **Risk Level** | **Consequence** | **Prevention/Mitigation** | **Contingency/Recovery** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Incomplete or unclear scope. | The project was barley documented. | 1 | 4 | 4 | Project not completed and delivered as required by client. | Prepare a clear listing of functions, features and components. Prepare a prototype. | Negotiate variation(s) with client if/where applicable. Bring on additional resource(s) as required. |
| 2 | Data lost. | The data has been corrupted. | 7 | 8 | 56 | Teams can’t keep track of bugs in their projects. | * Allow database to backup locally and or to other remote locations. * Do not store database files on storage media like a USB or external HDD. * Do not store database files in compressed folder. * Anti-virus software is installed on server machine that the data will be backed up to. | * Use database console commands to restore database. * Use third party tools that are designed to recover data from that specific database. * Use error logs for application database to diagnose and determine cause of problems. |
| 3 | Technical Debt. |  | 2 | 2 | 4 | Project not completed and delivered as required by client or client discovers bugs. | * Ensure technical documentation about project is emailed to QA team before developers so they can get to work in preparing a test plan. * Use Continuous Integration. | * Use a dedicated sprint to address a large amount of technical debt * Use sprint by sprint to reduce tech debt gradually. |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

***Note****: Risk Level = Likelihood x Impact*

## Sources

How to Repair MDF Files in SQL Server Database

<https://dzone.com/articles/how-to-repair-mdf-file-in-sql-server-database>