CSC3003S Capstone Project — Stage One

Risk [30 Marks]

|  |  |  |  |
| --- | --- | --- | --- |
| Project Abbrev & Name | ViKER Interface | Client/Supervisor + email | Maria Keet  [mkeet@cs.uct.ac.za](mailto:mkeet@cs.uct.ac.za) |
| Date | 22 Jul 2019 | Tutor + email | Ryan Lazar  [lzrrya001@myuct.ac.za](mailto:lzrrya001@myuct.ac.za) |
| Team Members | Jeremy Du Plessis | Gabriel Stein | St John Grimbly |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Risk Condition [1]** | **Consequence [1]** | **Cat** | **Prob [½]** | **Impact [½]** | **Mitigation [1]** | **Monitoring [1]** | **Management [1]** |
| A misunderstanding of the core project problem, due to the fact that it is partly theoretical and novel. | Time wasted planning and developing solutions to an incorrectly understood problem. | Development | Low | Critical | Allocate a greater proportion of time in the beginning of the project to understanding the problem by reading the necessary papers and theory. | Failed attempts to begin implementing the solution will indicate we have not understood the problem sufficiently well. | If we are unable to properly understand the problem after reading the papers and doing our own research, we will schedule a meeting with the client (Dr Keet) and bring questions for clarification. |
| Implementation of front-end interface using unknown packages may prove to be limiting or have foreseen negative effects (make extra work) | Time wasted and in the worst case limiting the functionality of the program. | Development | Low | Critical | Research several options for graphical interface libraries before selecting one. | More time than allocated spent trying to overcome unforeseen problems with the UI packages will be an indicator. | If we encounter excessive problems, we will revert to using a textual interface and focus on the implementation of the backend, returning to focusing on the UI once the back end is complete. |
| Since the transformation rules from ARM to the conceptual model is theoretical, there may be several edge cases which cause errors which may not be anticipated or tested. | A program which does not account for every possible error that may arise from transformations - which may be considered a bug. | Development | Medium | Critical | Prioritise the generation of test cases before starting any development, generating as many as possible up front and checking them with the client to ensure the I/O that we are testing is correct and that we have not missed any edge cases. | The only possible indicator for this risk is failed test cases. So, the only way to know would be to generate as many test cases as possible. | Failed test cases and exceptions are expected since this work has not been done before, so the appropriate way to manage this risk would be clear communication with the client. |
| Too much time spent on developing the graphical interface (focusing on usability over functionality). | Less time spent on back end implementation, potentially leading to bugs and poor-quality code. | Development | Low | Critical | Dedicate developer time disproportionately to developing the logic, rules and test cases in the beginning of the project. Concentrating the majority of developer time on the graphical interface only towards the end after the implementation and testing of the backend has been done. | An indicator will be if a disproportionately large proportion of tasks, tracked via a task management board, concern front-end work (translating to time spent). | If the front end is taking too much time, the team should switch to focusing on the back-end implementation and use a text-based user interface instead, only coming back to work on the GUI once the backend is complete. In the worst case, we submit a text-based interface which is satisfactory according to the requirements. |
| Our conceptual model and UI not being easy enough to understand for non-expert users. In other words, there is a risk of not creating an interface which solves the root issue of only experienced users of the database know how to query it. | Not making it easier for the end user to query the database without the need for expert knowledge of the underlying system. | Development | Medium | Critical | Use known UX and UI principles to create a user-friendly interface.  Test the interface and program with non-expert users and track their experience and feedback. | Responses from non-expert users in user-testing conducted as the project progresses (the client should also test out the prototype when it is ready and at subsequent meetings). If there is confusion on the part of the non-expert user during testing, we will know our UI is not performing it’s intended function. | Prototype several options / variations of the UI for user testing and take guidance from the responses as to how to alter the UI. |
| Improper team coordination and management. | Wasted time and resources. Potentially not finishing critical tasks in time. | Development | Medium | Critical | Create specific roles and make these roles clear to each team member. Invoke iterative development and have ‘daily standups’ to check in on daily progress. | Using weekly goals and checkpoints to measure progress. | If things are not being managed properly, mapping out detailed time planning with all team members will be essential to keep the project on track. |