CSC3003S Capstone Project — Stage One

Risk [30 Marks]

|  |  |  |  |
| --- | --- | --- | --- |
| Project Abbrev & Name | ViKER Interface | Client/Supervisor + email | Maria Keet + mkeet@cs.uct.ac.za |
| Date | 22 Jul 2019 | Tutor + email | Ryan Lazar + 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 whole problem, due to the fact that it is partly theoretical, may cause delays | Not enough time for implementation of final solution |  |  |  | Allocate a greater proportion of time to understanding the problem by reading the necessary papers etc. up front. |  |  |
| Implementation of front end interface using unknown packages that may be limiting or have foreseen negative effects (make extra work) | Time wasted towards the end of the project which may cause the quality of the user interface to suffer |  |  |  | Research several options for graphical interface libraries as well as have a plan for a textual interface as back up |  |  |
| 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 |  |  |  | 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. |  |  |
| Too much time spent on developing the graphical interface, not enough time developing and testing the software implementation of the transformation rules |  |  |  |  | Divide developer time disproportionately in the beginning of the project to work on the logic, rules and test cases and implement the software. 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. |  |  |
| 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. |  |  |  | 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. |  |  |
|  |  |  |  |  |  |  |  |

### Instructions

1. Complete the table with **realistic** risks to get marks.
2. You need at least 5 risks, you may have more.
3. Remove these instructions from your submission on Vula.

### Follow-up

1. Bring this document to your next meeting with your client (or email it beforehand) and ask if it adequately accounts for possible issues.
2. Resolve and note the solutions to any issues.