Agile Project Planning and Risk Management

Trung Le

Grantham University

**Agile Project Planning and Risk Management**

|  |  |  |
| --- | --- | --- |
| Iteration 1, 4 wks 9/1/20-9/29/20 | Iteration 2, 4 wks 9/30/20-10/28/20 | Iteration 3, 4 wks 10/29/20-11/25/20 |
| UC01: Search for Programs points: 5 priority: 1 depends on: None | UC02: Display Program Detail points: 3 priority: 2 depends on: UC01 | UC07: Check Application Status points: 3 priority: 2 depends on: UC06 |
| UC03: Submit Online Application points: 5 priority: 1 depends on: None | UC05: Logout points: 1 priority: 4 depends on: UC04 | UC08: Submit Recommendation points: 4 priority: 3 depends on: UC06 |
| UC04: Login points: 2 priority: 4 depends on: None | UC06: Edit Online Application points: 4 priority: 1 depends on: UC03 | UC09: Approve Course Equivalency points: 5 priority: 1 depends on: UC06 |

The three iterations of 4 weeks each are enough to complete the project. The toughest part would be the start as that consists of the foundation and other use cases depend on its success. Iterations one consists of UC01, UC03, and UC04 adding up to 288 total hours, well within the 480 available. 12 points are also well within the 3 points per week that the team can complete. Iterations 2’s UC all rely on previous iterations to complete to start, but the work is much lighter. Since iteration 2 can be completed much faster, that allows iteration 1 extra time if needed, and iterations 3 a head start. The three months should be ample time for the completion of the project.

|  |  |  |
| --- | --- | --- |
| **Risk** | **Risk Resolution Measures** | **Justification** |
| Staff leaving project | More talent staffing and cross-training | Many reasons for a team member to be no longer available to work on the project. PM should be prepared for that scenario. |
| Gold-plating | Requirements scrubbing and prototyping. | Adding extra features to a website is easy, but can get out of hand if not careful. Prototype and making sure all requirements are met and are a priority. |
| Continuous new requirements | Deferring new changes to later increments, and increasing change threshold | New requests for features will always come in, but it doesn't have to be implemented or included in initial requirements. Only top priority ones with approval should be considered. |
| Real-time performance shortfall | Test and benchmark. Budget for hardware upgrade if necessary. | Bad or unoptimized code can lead to poor performance along with inadequate server hardware. |
| Difficult API usage | Integrate standard and commonly available API, be prepared to write a new one. | Many open-source API available to use, but some might be difficult to use and integrate. The project assumes those can be used. |

**References**

Kung, David. (2014). Object-Oriented Software Engineering: An Agile Unified Methodology. McGraw-Hill Higher Education