Resource Manager

Iteration Plan

# 1. Key milestones

|  |  |
| --- | --- |
| **Milestone** | **Date** |
| Iteration start | 15th April 2019 |
|  |  |
|  |  |
|  |  |
| Planning for the remaining semester completed | 28 April 2019 |
| Iteration stop | 29th April 2019 |

# 2. High-level objectives

1. Estimate project scope
2. understand how the architecture meets non functional requirements

# 3. Evaluation criteria

1. The team understand the scope of the project
2. The team understands how the application architecture meets the functional requirements

# 4. Work Item assignments

The following Work Items will be addressed in this iteration:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Work Item ID** | **Name or key words of description** | **Outcome** | **State** | **Assigned to (name)** | **Estimated Hours** | **Hours worked** | **Estimate of hours remaining** |
| 1.1 | Estimate times for Future tasks |  | Complete | Aaron | 2 | 1.5 | 0 |
| 1.2 | Estimate times for Future tasks |  |  | Brodie | 2 |  |  |
| 1.3 | Estimate times for Future tasks |  |  | Jack | 2 |  |  |
| 1.4 | Estimate times for Future tasks |  |  | Matt | 2 |  |  |
| 2.1 | Brief description of how architecture meets non functional requirements |  |  | Aaron | 2 | 2 |  |
| 2.2 | Brief description of how architecture meets non functional requirements |  |  | Brodie | 2 |  |  |
| 2.3 | Brief description of how architecture meets non functional requirements |  |  | Jack | 2 |  |  |
| 2.4 | Brief description of how architecture meets non functional requirements |  |  | Matt | 2 |  |  |
| 3.1 | Create iteration plan 4 |  |  | Online meeting as team | 2 |  |  |
| 3.2 | Update and finalize iteration plan 3 |  |  | Online meeting as team | .5 |  |  |
| 3.3 | Ensure leave is calculated in project calendar |  |  | Aaron | 1 | .5 | 0 |
| 3.4 | Ensure leave is calculated in project calendar |  |  | Brodie | 1 |  |  |
| 3.5 | Ensure leave is calculated in project calendar |  |  | Jack | 1 |  |  |
| 3.6 | Ensure leave is calculated in project calendar |  |  | Matt | 1 |  |  |

# 5. Issues

[List any issues to be solved during the iteration. Update status when new issues are reported during the daily meetings]

|  |  |  |
| --- | --- | --- |
| **Issue** | **Status** | **Notes** |
| BN |  |  |

# 6. Assessment

[Use this section for capturing and communicating results and actions from assessments, which are typically done at the end of each iteration. If you don’t do this, the team may not be able to improve the way they develop software.]

|  |  |
| --- | --- |
| Assessment target | [This could be the entire iteration or just a specific component] |
| Assessment date |  |
| Participants |  |
| Project status | [For example, express as Red, Yellow, or Green.] |

## Assessment against objectives

[Document whether you addressed the objectives as specified in the Iteration Plan.]

## Work Items: Planned compared to actually completed

[Summarize whether all Work Items planned to be addressed in the iteration were addressed, and which Work Items were postponed or added.]

## Assessment against Evaluation Criteria Test results

[Document whether you met the evaluation criteria as specified in the Iteration Plan. This could include information such as “Demo for Department X was well-received, with some concerns raised around usability,” or “495 test cases were automated with a 98% pass rate. 9 test cases were deferred because the corresponding Work Items were postponed.”

The most important guideline here is that although satisfactorily completed items may summarily mentioned, incomplete items require a more comprehensive explanation ]

## Other concerns and deviations

[List other areas that have been evaluated, such as financials, or schedule deviation, as well as Stakeholder feedback not captured elsewhere.]