Master Test Plan

9/03/2018

# **Version Information**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Authors | Description |
| 0.1 |  | Michelle Vinall | Initial draft of Master Test Plan |
|  |  |  |  |
|  |  |  |  |

# **Distribution List**

|  |  |
| --- | --- |
| Name | Function |
| Aaron Peachey | Team Member |
| Charnes Nell | Team Member |
| Collin Mckeahnie | Team Member |
| Michelle Vinall | Team Member |

# **Management Summary**

|  |
| --- |
| Project Objective  The project objective is to create an online quiz game capable of single and multiplayer modes that have integration with Facebook and Google Play Services to allow features such as sharing, challenging, viewing leader boards and achievements. |
| Test Objective and Assignment  This test plan will implement the following objectives:   * To provide test coverage for 100% of the documented requirements * To provide a testing guide for the development of all functionality and performance testing required * To execute 100% of the test cases during User Acceptance Testing * To execute 100% of the test cases during Unit Testing * To execute 100% of the test cases during Integration Testing * Provide a schedule of all proposed tests   Assigned to   * Aaron Peachey-Tester * Charnes Nell-Tester * Collin Mckeahnie-Reviewer and Tester * Michelle Vinall-Author and Tester |
| Short Description of the Test Approach  The test approach of this document is to test the software user interface and functionality via several testing techniques. |
| Results to be Realised   |  |  |  | | --- | --- | --- | | **Result** | **Document** | **Delivery Date** | | Are users able to login | Unit testing report | 17 May 2018 | | Are users able to answer questions | Unit testing report | 17 May 2018 | | Is the game keeping the correct score | Unit testing report | 17 May 2018 | | Is the round timer working | Unit testing report | 17 May 2018 | | Are all buttons working as they should | UAT test report | 1 June 2018 | | Is text readable | UAT test report | 1 June 2018 | | Is data being stored correctly | UAT test report | 1 June 2018 | | Can users check global scores | UAT test report | 1 June 2018 | | Can users check leader board | UAT test report | 1 June 2018 | | Can users check achievements | UAT test report | 1 June 2018 | | Can users add questions | UAT test report | 1 June 2018 | | Do all components work together completely | Integration test report | 24 May 2018 | | Test flows of use cases | Integration test report | 24 May 2018 | | Is the user Interface easy to navigate | UAT Testing | 24 May 2018 | |
| Qualitative Objectives  There will be deadlines for all testing procedures. All results will be validated and an end report will document all results prior to 4 June |
| Estimated Completion  4 June 2018 |
| Test process risks and measures   |  |  | | --- | --- | | *Test process risks*   * Testing data/ environment not available * Communication problems * Delay in starting testing * Natural disasters * Undefined requirements * Inadequate testing with defects found at a late stage | *Measures to be taken*   * Prepare testing data and environments well ahead of time according the test plan * Make sure all documentation is well maintained and distributed to all team members * Add a buffer to schedule to allow for unexpected contingencies * Provide a back-up tester for all tests in case of unforeseen circumstances * Make a requirements list that is update regularly during the planning stage * Organize a thorough testing schedule, that outlines all tests and testing order | |
| Go/no-go decisions   * After each test level the test manager makes sure that a test report is drawn up. * If this is signed off on the next level can be started |

Table of Contents

[**Version Information** 1](#_Toc510087813)

[**Distribution List** 1](#_Toc510087814)

[**Management Summary** 1](#_Toc510087815)

[1. Introduction 5](#_Toc510087816)

[1.1 Project Objective 5](#_Toc510087817)

[1.2 Objective 5](#_Toc510087818)

[1.3 Involved in Creating the Master Test Plan 5](#_Toc510087819)

[2. Assignment Formulation 5](#_Toc510087820)

[2.1 Client 5](#_Toc510087821)

[2.2 Supplier 5](#_Toc510087822)

[2.3 Assignment 5](#_Toc510087823)

[2.4 Scope 5](#_Toc510087824)

[2.4.1 Functions to be tested. 5](#_Toc510087825)

[2.4.2 Functions not to be tested. 6](#_Toc510087826)

[2.5 Preconditions and Assumptions 6](#_Toc510087827)

[2.6 Acceptors and Acceptance Criteria 6](#_Toc510087828)

[2.6.1 Acceptants 6](#_Toc510087829)

[2.6.2 Acceptation criteria 6](#_Toc510087830)

[3. Documentation 7](#_Toc510087831)

[3.1 Basis for the Master Test Plan 7](#_Toc510087832)

[3.2 Standards 7](#_Toc510087833)

[3.3 Test basis 7](#_Toc510087834)

[4. Test Strategy 7](#_Toc510087835)

[4.1 Product Risk Analysis 7](#_Toc510087836)

[5. Approach 8](#_Toc510087837)

[5.1 Test Levels 8](#_Toc510087838)

[5.2 The Unit Testing 8](#_Toc510087839)

[5.2.1 Goal 8](#_Toc510087840)

[5.2.2 Short Description 8](#_Toc510087841)

[5.2.3 Responsible 8](#_Toc510087842)

[5.3 The Integration Testing 8](#_Toc510087843)

[5.3.1 Goal 8](#_Toc510087844)

[5.3.2 Short Description 8](#_Toc510087845)

[5.3.3 Responsible 8](#_Toc510087846)

[5.4 The User Acceptance Testing 8](#_Toc510087847)

[5.4.1 Goal 8](#_Toc510087848)

[5.4.2 Short Description 8](#_Toc510087849)

[5.4.3 Responsible 8](#_Toc510087850)

[5.5 Phasing Per Test Level 8](#_Toc510087851)

[5.6 Entrance and Exit Criteria for Each Test Level 9](#_Toc510087852)

[5.6.1 Functional Acceptance Test 9](#_Toc510087853)

[5.6.2 User Acceptance Test 9](#_Toc510087854)

[6. Organization 9](#_Toc510087855)

[6.1 Organization structure 9](#_Toc510087856)

[6.2 Roles, tasks and responsibilities 9](#_Toc510087857)

[6.3 Structure of meetings 10](#_Toc510087858)

[6.4 Structure of reporting 10](#_Toc510087859)

[6.5 Completion 10](#_Toc510087860)

[7. Infrastructure 10](#_Toc510087861)

[7.1 Test environments 10](#_Toc510087862)

[7.2 Test tools 10](#_Toc510087863)

[7.3 Office setup 10](#_Toc510087864)

[8. Management 10](#_Toc510087865)

[8.1 Test process management 10](#_Toc510087866)

[8.2 Test infrastructure management 10](#_Toc510087867)

[8.3 Test product management 10](#_Toc510087868)

[8.4 Defects procedure 10](#_Toc510087869)

[9. Test Process Risks and Countermeasures 10](#_Toc510087870)

[10. Global Estimation & Planning 10](#_Toc510087871)

[10.1 Estimation 10](#_Toc510087872)

[10.2 Planning 10](#_Toc510087873)

[10.3 Milestones 10](#_Toc510087874)

[11. Glossary 10](#_Toc510087875)

# 1. Introduction

## Project Objective

The project objective is to create an online quiz game capable of single and multiplayer modes that have integration with Facebook and Google Play Services to allow features such as sharing, challenging, viewing leader boards and achievements.

## 1.2 Objective

This test plan will implement the following objectives:

* To provide test coverage for 100% of the documented requirements
* To provide a testing guide for the development of all Functionality and Performance Testing required
* To execute 100% of the test cases during Unit Testing
* To execute 100% of the test cases during Integration Testing
* To execute 100% of the test cases during User Acceptance Testing
* Provide a schedule of all proposed tests

## 1.3 Involved in Creating the Master Test Plan

|  |  |  |
| --- | --- | --- |
| Name | Function | Responsibility |
| Michelle Vinall | Team Member | Writing the document |
| Collin Mckeahnie | Team Member | Reviewer |
| Aaron Peachey | Team Member | Global Reviewer |
| Charnes Nell | Team Member | Global Reviewer |

# 2. Assignment Formulation

## 2.1 Client

CSU Lecturer Jim Tulip

## 2.2 Supplier

Online Question Game Group

## 2.3 Assignment

The client wants an on-line Quiz game with multi and single player mode that can integrate with Facebook and Google Play Services. The tests outlined will be testing the functional and non-functional requirements set out in in the Initial Requirements Model document. The results from these test should show full use case realisation.

## 2.4 Scope

### 2.4.1 Functions to be tested.

* Testing the gaming environment
* Verify all game elements do what is required of them
* User interface
* User interactions
* Menus and their functions
* Readable text
* Data integrity testing
* Game flow
* Scoring and timing
* Multi-player testing/load testing
* Testing for integration and compatibility of external SDKs/databases
* Server to database connection

### 2.4.2 Functions not to be tested.

* Localization testing will not be performed in this version
* Actual testing of SDK components as these are externally tested by the companies supplying the packages

## 2.5 Preconditions and Assumptions

Preconditions

* Test plan must be completed by 9th of April.
* Testing must be completed by 4th of April.
* The prevailing project plan is boundary for this master test plan and the execution of the test project is based on this.

Assumptions

* The application will run on Android and IOS operating systems
* Standard input will be keyboard and touch

## 2.6 Acceptors and Acceptance Criteria

### 2.6.1 Acceptants

The table below states the acceptants of the Let’s Quiz application

|  |  |  |
| --- | --- | --- |
| Name | Function | Department |
| Jim Tulip | Lecturer | IT303 Lecturer CSU |
| The Client | Sign off and project acceptance | Online Question Game Group |
|  |  |  |

### 2.6.2 Acceptation criteria

The table below states which acceptance criteria there are for the Let’s Quiz application and to which standard they should apply:

|  |  |
| --- | --- |
| Description | Standard |
| Game must be playable on different operating systems. | High |
| The game should allow 1/multiple players. | High |
| The game must display a login scene to the user. | High |
| Menu screen must allow player to start a game. | High |
| Menu screen should allow players to invite, share and like on Facebook. | Medium |
| Menu screen must allow player to enter settings menu. | Medium |
| Menu screen must allow player to access high scores. | Medium |
| Questions must be presented to the player. | High |
| User must be able to choose an answer. | High |
| The application must be able to determine if the player has selected the correct answer. | High |
| The score of each player must be recorded. | High |
| The application must have the ability to determine the winner. | High |
| A congratulatory message should be displayed to the winning player. | Medium |
| A list of correct answers will be displayed to the user when the game is over. | Medium |
| When the game ends the program should ask the player if a new game should be started. | High |
| A timer should limit the amount of time of each round. | High |
| Application must meet the subject outline criteria | High |

# 3. Documentation

## 3.1 Basis for the Master Test Plan

The following documents are used as basis for this master test plan.

|  |  |  |  |
| --- | --- | --- | --- |
| **Document name** | **Version** | **Date** | **Author** |
| Initial Requirements Model | 0.3 |  | Michelle Vinall |
| Project Plan | 0.1 |  | Charnes Nell |

## 3.2 Standards

The following conventions and standards are applied for this test plan.

|  |  |  |  |
| --- | --- | --- | --- |
| **Document name** | **Version** | **Date** | **Author** |
| C# Coding Convensions | NA | 20 July 2015 | [https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions#naming-conventions](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions%23naming-conventions) |
| PHP Coding Standards | NA | 31 March, 2008 | PSR-1 |
| SQL Server Database Coding Standards and Guidelines | 1.0 | NA | SQLAuthority |

## 3.3 Test basis

The test basis contains the documentation that serves as basis for the tests that have to be executed. The overview below describes the documentation that is the starting point for testing.

|  |  |  |  |
| --- | --- | --- | --- |
| **Document name** | **Version** | **Date** | **Author** |
| Master Test Plan | 0.1 |  | Michelle Vinall |
| Initial Requirement Model | 0.2 |  | Michelle Vinall |
| Project Plan | 0.1 |  | Charnes Nell |

# 4. Test Strategy

## 4.1 Product Risk Analysis

During the risk assessment the test goals were also formulated. These can be found together with the corresponding characteristics in table below.

|  |  |  |
| --- | --- | --- |
| Test goal | Description | Characteristic |
| Functionality | To ensure the application works as intended | Game progress, performance, basic usability, error conditions, accessibility |
| OS compatibility | To ensure game works in all intended operating systems | Compatibility with android and IOS |
| Data integrity | Verify the data stored in the database is accurate and reliable | Data correctness |

# 5. Approach

## 5.1 Test Levels

For this MTP the following test levels are acknowledged:

|  |  |
| --- | --- |
| Test Level | Goal |
| Unit Testing | To isolate each part of the program and test that the individual parts are working correctly |
| Integration Testing | To test the interaction between individual modules. |
| User Acceptance Testing | Testing of the application in a real environment |

## 

## 5.2 The Unit Testing

### 5.2.1 Goal

To isolate each part of the program and test that the individual parts are working correctly

### 5.2.2 Short Description

The code will be broken into units and individually tested using NUnit, to see that they do their required job.

### 5.2.3 Responsible

Aaron Peachey

Charnes Nell

Collin Mckeahnie

Michelle Vinall

## 5.3 The Integration Testing

### 5.3.1 Goal

To test the interaction between individual software units

### 5.3.2 Short Description

Each software unit is integrated one by one to test if they work in combination with each other, to test whether the required implementation exists between modules.

### 5.3.3 Responsible

Aaron Peachey

Charnes Nell

Collin Mckeahnie

Michelle Vinall

## 5.4 The User Acceptance Testing

### 5.4.1 Goal

Testing of the application in a real environment.

### 5.4.2 Short Description

This is where the software application is tested by running it in the environment it is intended for and check if it meets the business requirements.

### 5.4.3 Responsible

Aaron Peachey

Charnes Nell

Collin Mckeahnie

Michelle Vinall

## 5.5 Phasing Per Test Level

Phases to plan, execute and complete testing of each testing level

* Analysis of testing requirements
* Planning of tests to be used
* Development of tests
* Execution of tests
* Reporting of test results
* Analysing test results
* Documentation of test results
* Test level closure

## 5.6 Entrance and Exit Criteria for Each Test Level

### 5.6.1 Functional Acceptance Test

Unit testing entrance criteria

* Initial requirements defined
* Master test plan
* Test cases defined
* Testable code with an appropriate test environment is available

Unit testing exit criteria

* Identify all defects and fix them to an acceptable standard
* Ensure that all critical test cases have been completed and passed
* All test reports are completed and signed off

Integration testing entrance criteria

* Successful completion of all unit tests
* Integration test cases and test environment is available.

Integration testing exit criteria

* Identify all defects and fix them to an acceptable standard
* Ensure that all critical test cases have been completed and passed
* All test reports are completed and signed off

### 5.6.2 User Acceptance Test

UAT testing entrance criteria

* Functional and initial requirements have been met
* Successful completion of all integration tests
* UAT test cases and environment are available

UAT testing exit criteria

* Identify all defects and fix them to an acceptable standard
* Ensure that all critical test cases have been completed and passed
* All initial requirements were fulfilled
* All test reports are completed and signed off.

# 6. Organization

## 6.1 Organization structure

The team consists of four members and one assessor:

Aaron Peachey Team Member

Charnes Nell Team Member

Collin Mckeahnie Team Member

Michelle Vinall Team Member

Jim Tulip Assessor

## 6.2 Roles, tasks and responsibilities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Role** | **Department / Name employee(s)** | **# hours**  **per week** | **Period** | **Description of tasks and**  **responsibilities** |
| Test manager | Michelle Vinall | 10-12 | 4 weeks | Write MTP  Coordinate overall test process |
| Test coordinator | Aaron Peachey  Charnes Nell  Collin Mckeahnie  Michelle Vinall | 10-12 | 3 weeks | Write test plans  Coordinate tests |
| Tester | Aaron Peachey  Charnes Nell  Collin Mckeahnie  Michelle Vinall | 10-12 | 4 weeks | Make test specifications  Execute (re)tests |

## 6.3 Structure of meetings

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Goal** | **Frequency** | **Who** |
| Project meeting | Discuss overall project progress including decision making and group processes | Weekly | Team |
| Status Update meeting for each test level | Discuss progress including problem solving, prioritization for each test level | Weekly | Team Testers |
| Defect triage | Discuss and prioritize defects found during the different testing levels | Weekly | Team Testers |

## 6.4 Structure of reporting

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Goal** | **Frequency** | **Who** |
| Risk report | To outline the established risks, their priority and mitigation strategies we will use | When necessary | Aaron Peachey  Charnes Nell  Collin Mckeahnie  Michelle Vinall |
| Release advice | Discuss the test pricedures and when and how they will be executed | Once-only | Aaron Peachey  Charnes Nell  Collin Mckeahnie  Michelle Vinall |
| Status Report | This will provide a summary of the overall test case status | Weekly | Aaron Peachey  Charnes Nell  Collin Mckeahnie  Michelle Vinall |
| Defect status reports | To notify the team of any defects and their status | Weekly | Aaron Peachey  Charnes Nell  Collin Mckeahnie  Michelle Vinall |
| Test Summary Report | This document is to explain various details and activities about the testing performed on the software application. | Once only | Aaron Peachey  Charnes Nell  Collin Mckeahnie  Michelle Vinall |

## 6.5 Completion

To sign off on completion we will have a playable quiz game that meets all requirement criteria, with none to minimal defects.

# 7. Infrastructure

## 7.1 Test environments

| **Test level** | **Environment** | **Requirements** | **From** | **To** |
| --- | --- | --- | --- | --- |
| UT | * NUnit/Mockito * Mono Develop/ Visual Studio * GitHub * Unity3D | * Requirements must be available * The environment will be needed for two weeks till unit testing is complete. * Readable test case * Repeatable tests * Testers * Progress and end of test reports | 4 May 2018 | 18 May 2018 |
| IT | * NUnit/Mockito * Mono Develop/ Visual Studio * GitHub * Unity3D | * Requirements must be available * The environment will be needed for one week till integration testing is complete. * Readable test case * Repeatable tests * Unit testing should be completed * Testers * Progress and end of test reports | 18 May 2018 | 25 May 2018 |
| UAT | * Running version of the game on Android and IOS | * Requirements must be available * Test cases * Integration testing should be completed * Testers * Progress and end of test reports | 25 May 2018 | 1 June 2018 |

## 7.2 Test tools

|  |  |  |
| --- | --- | --- |
| **Test level** | **Test tool** | **Comment** |
| UT | Personal computer OS Win 7 upwards  NUnit  Mockito  GitHub repository  Mono develop/ Visual Studio  SQL Server  Let’s Quiz application | These will be using NUnit, Mockito testing tools as well as NUnit assert.  The tools will set up and communally used through the GitHub repository.  Test tools can execute tests scripts much more reliably that humans. |
| IT | Personal computer OS Win 7 upwards  NUnit  Mockito  GitHub repository  Mono develop/ Visual Studio  SQL Server  Let’s Quiz application | As above |
| UAT | Mobile device running Android and IOS  Let’s Quiz application | The application will be installed on the before mentioned mobile devices or emulators if no mobile device is usable. |

## 7.3 Office setup

|  |  |  |
| --- | --- | --- |
| **Test level** | **Components** | **Comment** |
| UT | Software components, local and remote communal repository (GitHub)/test server, MS Word and MS Excel | Access to these will be through the communal repository and local machines |
| IT | Software components, local and remote communal repository (GitHub)/ test server, MS Word and MS Excel | Access to these will be through the communal repository and local machines |
| UAT | Application server/Front end running environment, hardware server, database server, network, local and remote communal repository (GitHub), MS Word and MS Excel | This will be accessed through the application itself and the remote GitHub repository. |

# 8. Management

## 8.1 Test process management

The management of the test process can be divided into three parts:

* Progress and expenditure of budget and time: the management of the planning and guarding of the progress in terms of time, resources and means. This has been arranged as followed:
  + Time of project stage 1 -14 weeks
    - Testing time- 7 weeks all levels including test planning
    - Actual testing time- 4 weeks
    - Planning tests and setting up test cases- 3 weeks
  + Costs-Testing phase-
    - Total testing costs – 35% of overal budget
    - 4 testers-1 week $3,520 - total- $24,640
  + Time of project stage 2 -14 weeks
    - Testing time- 5 weeks all levels including test planning
    - Actual testing time- 2 weeks
    - Planning tests and setting up test cases- 3 weeks
  + Costs-Testing phase-
    - Total testing costs – 35% of overal budget
    - 4 testers-1 week $3,520 - total- $17,600
* Quality indicators: the aim of testing is to provide information and advice on the risks and quality of the object to be tested. To be able to provide this information, quality indicators are registered. This has been arranged as followed:
  + Progress/yield – amount of testing done by each developer per level
  + Covered requirements- requirements covered per test
  + Stability-assess whether the tested unit is stable enough to progress to the next test level
  + Test coverage – amount of the software system tested
  + Efficency of defect detection- amout of defects detected per test level
  + Defect removal – amount of defects fixed per test level
  + Passed requirements-amount of requirements checked
  + Tests passed- amout of tests pased
* Test statistics: the test manager builds statistics based on the above information. Statistics can supply insight into the progress of the test process and quality of the test object, including any trends. This has been arranged as followed:
  + Statistics-Tracking
    - Passed test cases percentage
    - Failed test cases percentage
    - Fixed Defects percentage
    - Average time to repair defects
    - First run fail rate
  + Statistics – Effort
    - Number of tests run during time period
    - Test design efficiency
    - Test review efficiency
    - Ecuted pecentage of tests run

## 8.2 Test infrastructure management

* Procedure office setup
  + How Managed-
    - Maintenance of Teams Github repository
    - Make sure every one has access to the required office setup equipment
    - Look into any issues of the team office setup
    - Co-ordination till an issue is resolved
  + By Whom- Aaron and Collin
* Procedure test tool
  + How Managed-
    - Maintenance of teams Github repository
    - Set up uf the requred tools in the teams Github repository
    - Look into any issues found with the testing tools
    - Co-ordination till an issue is resolved
  + By Whom- Charnes and Michelle
* Procedure test environment
  + How Managed-
    - Maintenance of Teams Github repository with all the updated versions of test environments.
    - Updating/deleting outdated test-environments
    - Look into any issues on the environment
    - Co-ordination till an issue is resolved.
  + By Whom-Aaron. Charnes, Collin and Michelle

## 8.3 Test product management

* All test cases and project documents will be uploaded to version control in a project repository.
* All documents will take in a standardized form and style
* All tests will be performed on the local repository on the individual testers computer then will be pushed to the shared remote repository for version control and storage

## 8.4 Defects procedure

For the registration and maintenance of defects the following tool is being used: Defects and Maintenance Register

The responsibility for the observance of this defects procedure lies with the defect administrator: Michelle Vinall.

# 9. Test Process Risks and Countermeasures

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Number | Event | | Consequence | Impact | Chance | Score | Countermeasures | Owner |
| 01 | | Testing data/ environment not available | Requirements are missing, incomplete or incorrect and cause a delay | 3 | 1 |  | Prepare testing data and environments well ahead of time according the test plan | Aaron  Michelle |
| 02 | | Communication problems | This can occur when test documents are not maintained or inadequate communication concerning testing is given to team members. | 3 | 2 |  | Make sure all documentation is well maintained and distributed to all team members | Charnes  Aaron |
| 03 | | Delay in starting testing | The schedule of testing is often inadequate for the amount of testing that should be performed. | 3 | 3 |  | Add a buffer to schedule to allow for unexpected contingencies | Collin  Charnes |
| 04 | | Natural disasters | Team member not available due to unforeseen reason | 2 | 1 |  | Provide a back-up tester for all tests in case of unforeseen circumstances | Michelle  Collin |
| 05 | | Undefined requirements | If requirements are unknown or incorrect this can cause confusion when testing causing a delay | 3 | 2 |  | Make a requirements list that is update regularly during the planning stage | Aaron  Charnes  Collin  Michelle |
| 06 | | Inadequate testing with defects found at a late stage | These defects can be time consuming to fix | 3 | 2 |  | Organise a thorough testing schedule, that outlines all tests and testing order | Aaron  Charnes  Collin  Michelle |

Impact:

* 3 - High
* 2 - Medium
* 1 - Low

Chance:

* 3 - Frequent
* 2 - Occasional
* 1 – Unlikely

Risk Matrix – Score:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Impact Severity | | |
| 1 | 2 | 3 |
| Chance Factor | 3 | Moderate | High | High |
| 2 | Low | Moderate | High |
| 1 | Low | Low | Moderate |

# 10. Global Estimation & Planning

## 10.1 Estimation

The estimation is as follows: << The estimation divided in personal and infrastructural costs. >>

As a guide we are using this break down for effort:

* Requirements Analysis-35%
* Preliminary and Detailed design-30%
* All testing procedures-35%

| **Test level** | **Who** | PC |  |  |  |  |  |  | **Totals** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Overall | Test manager |  |  |  |  |  |  |  |  |
| UT | Test coordinator |  |  |  |  |  |  |  |  |
|  | Test specialists |  |  |  |  |  |  |  |  |
| IT | Test coordinator |  |  |  |  |  |  |  |  |
|  | Test specialists |  |  |  |  |  |  |  |  |
| UAT | Test coordinator |  |  |  |  |  |  |  |  |
|  | End users |  |  |  |  |  |  |  |  |
|  | Test specialists |  |  |  |  |  |  |  |  |
| **Totals:** | |  |  |  |  |  |  |  |  |

## 10.2 Planning

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Phasing and activities | Week | | | | | | | | | | | | |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | | 14 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |
| Planning |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |
| Master test plan |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |
| Infrastructure |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |
| Unit test |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |
| Unit test plan |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |
| Specification |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |
| Execution |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |
| Conclusion |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |
| Integration Test |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |
| Integration test plan |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |
| Specification |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |
| Execution |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |
| Conclusion |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |
| User Acceptance Test |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |
| UAT test plan |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |
| Specification |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |
| Execution |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |
| Conclusion |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |
| Sign off on all testing |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |

## 10.3 Milestones

|  |  |
| --- | --- |
| Milestone description | Date |
| Finish planning | 12 March 2018 |
| Sign of on Master Test plan | 8 April 2018 |
| Conclude Unit Tests | 17 May 2018 |
| Conclude Integration testing | 24 May 2018 |
| Conclude UAT testing | 1 June 2018 |
| Sign off on all testing | **3 June 2018** |

# 11. Glossary

|  |  |
| --- | --- |
| IT | Integration testing- Testing to check that the different units of scan work together |
| MTP | Master Test Plan-A document that sets out a guide to the software testing process of the application |
| IOS | I operating system- the operating system employed by apple |
| OS | Operating system- The system software that manages a computer |
| UAT | User Acceptance testing- A testing process that tests the application in a real world environment |
| UT | Unit testing- the testing if the individual unis of the software |