**Group members**

Dang Thi Thao My - 15905067

Bui Nguyen Thien Khanh - 15907679

Nguyen Anh Quan - 15905072

Le Huynh Anh Tuan - 1325907

**Supervisor:** MSc. Truong Phuoc Loc

**Product Owner:** Mr. To Hoa Duy Man

**Test plan**

**Performance testing dashboard**

Version 1.6 – Released date: 05/07/2017

**Status**:

Approved by:

Released by: Capstone team (ETI1)

Internal

Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Changed by | Modifications |
| 1.0 | 01.05.2017 | Team | Initial version |
| 1.1 | 01.11.2017 | Team | Upload to OneDrive |
| 1.2 | 01.16.2017 | Team | Add new module |
| 1.3 | 03.01.2017 | Team | Fixing test strategy |
| 1.4 | 03.21.2017 | Team | Modify test date |
| 1.5 | 05.01.2017 | Quan Nguyen | Add source and reference |
| 1.6 | 05.07.2017 | Quan Nguyen | Change Template |

List of Abbreviations

CCR **C**ritical **C**omputer **R**esources  
CI **C**onfiguration **I**tem  
CM **C**onfiguration **M**anagement  
HLD **H**igh **L**evel **D**esign

RBVH  **R**obert **B**osch **E**ngineering and Business Solutions Limited

RADAR **R**eview **A**nd **D**efect **A**nalyze**r**

SW **S**oftware

V&V **V**erification and **V**alidation

Contents

[1. Introduction 1](#_Toc482444575)

[1.1. TEST SCOPES 1](#_Toc482444576)

[1.2. OUT OF SCOPES 1](#_Toc482444577)

[1.3. REFERENCES 1](#_Toc482444578)

[2. Test Approach and Test Methods 2](#_Toc482444579)

[2.1. TEST APPROACH 2](#_Toc482444580)

[2.2. TEST METHODS 2](#_Toc482444581)

[3. Test Scenario 3](#_Toc482444582)

[3.1. TEST ITEMS AND IDENTIFIERS 3](#_Toc482444583)

[3.2. FEATURES TO BE TESTED 3](#_Toc482444584)

[3.3. FEATURES NOT TO BE TESTED 4](#_Toc482444585)

[3.4. TEST OBJECTIVES 4](#_Toc482444586)

[3.5. TEST ENTRY CRITERIA 4](#_Toc482444587)

[3.6. TEST EXIT CRITERIA 4](#_Toc482444588)

[4. Testing levels 5](#_Toc482444589)

[4.1. UNIT TESTING 5](#_Toc482444590)

[4.2. SANITY TESTING 5](#_Toc482444591)

[4.3. SYSTEM TESTING 5](#_Toc482444592)

[4.4. AD-HOC/RANDOM TESTING 5](#_Toc482444593)

[4.5. RESOURCES AND BUDGET 6](#_Toc482444594)

[5. Risk list 8](#_Toc482444595)

[6. Test case/report templates 9](#_Toc482444596)

[7. Test environment and implementation 10](#_Toc482444597)

[7.1. TEST ENVIRONMENT 10](#_Toc482444598)

[7.2. TEST IMPLEMENTATION AND EXECUTION 10](#_Toc482444599)

[8. Suspension criteria and resumption testing 11](#_Toc482444600)

[9. Test deliverables 12](#_Toc482444601)

[10. Test planning, estimates and control 13](#_Toc482444602)

[11. Acceptance criteria 14](#_Toc482444603)

# Introduction

## TEST SCOPES

The purpose of the Test Plan is to gather information necessary to plan and control the test work in testing of the Performance Testing Dashboard project. It demonstrate and describe the approach and technique will be implemented to record and following up the test activities

The Test Plan for Performance Testing Dashboard support the following objectives:

* Implementation of Test Process
* List the Test Requirements
* Test Execution for the test cases derived from the modules.
* Outline the testing strategies to be employed
* Identify the required resources and provide an estimation of the test efforts.
* List the deliverable items of the test process.

## OUT OF SCOPES

Any tasks have not mentioned as above.

## REFERENCES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **REFERENCES** | | | | | |
| No. | Title | Version | Author | Date | Source |
| 1 | SRS | 1.0 | Scrum Team | 10.17.2016 | Docupedia |
| 2 | High Level Design | 1.0 | Scrum Team | 10.17.2016 | Docupedia |
| 3 | Low Level Design | 1.0 | Scrum Team | 10.17.2016 | Docupedia |
| 4 | Test Specification Report | 1.0 | Scrum Team | 10.11.2016 | Docupedia |
| 5 | Source Code | 1.0 | Scrum Team | 1.5.2017 | Docupedia |
| 6 | Issue Log | 1.0 | Scrum Team | 1.5.2017 | Docupedia |

# Test Approach and Test Methods

## TEST APPROACH

The diagram below outlines the Test approach that will be followed:

**a.** Planning

**b.** Design/ Build Test

**c.** Execute Tests

**d.** Report

1. **Planning** involves creating Test Plan, Schedule, Test Approach, and define responsibilities.
2. **Design/Build Test** involves identifying Test Cases, Entry & Exit Criteria, Expected Results, etc. for System Testing. The activity involve in writing test cases for module right after the identification is done. In general, Test data/expected result will be identified by the member from SCRUM team and changed by cycle due to Agile Methodology.
3. **Execute Tests -** The tests designed in “Design/Build Tests” will be updated and executed. All results will be reviewed by another member of SCRUM Team, bugs and issues are registered will be filled out in Bug Report Form and given to the responsible developer. The product is eligible accepted when the critical test cases are passed on sanity testing,
4. **Report –** Report happens when all pre-defined exit criteria have been achieved.

## TEST METHODS

The correct implementation of the module/ modules shall be ensured by apply the test methods given below:

|  |
| --- |
| Methods 1 |
| Requirements based test |
| Fault injection test or negative testing |
| Ad-hoc testing  (Based on lessons learnt or expert knowledge) |
| Resource usage test(Some aspects can only be evaluated properly when software unit tests are executed on the target hardware or if an emulator for the target processor supports resource usage test) |

|  |
| --- |
| Methods |
| Analysis of functional safety requirements |
| Generation and analysis of equivalence classes |
| Evaluate the special characteristics identified |
| Analysis of boundary values |
| Error guessing (Based on lessons learned or expert knowledge) |

# Test Scenario

## TEST ITEMS AND IDENTIFIERS

The test items include the modules of the PTD software:

* User Management Module
* Project Management Module
* Test Management Module
* File Management Module
* Agent Management Module
* Report Management Module

Aspects of each module would be tested:

* Features and Implementations
* Interfaces
* Integration
* Usability

## FEATURES TO BE TESTED

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Key | Summary | Module |
| 1 | RE1-1 | Log in | User management |
| 2 | RE1-1 | Log out | User management |
| 3 | RE1-3 | View user profile | User management |
| 4 | RE1-4 | Edit user profile | User management |
| 5 | RE1-5 | Create new user | User management |
| 6 | RE1-6 | Delete user | User management |
| 7 | RE1-7 | Edit user account | User management |
| 8 | RE1-8 | View user list | User management |
| 9 | RE2-1 | View project list as a user | Project management |
| 10 | RE2-2 | View project list as an admin | Project management |
| 11 | RE2-3 | Create new project | Project management |
| 12 | RE2-4 | View project detail as a user | Project management |
| 13 | RE2-5 | View project detail as an admin | Project management |
| 14 | RE2-6 | Edit project | Project management |
| 15 | RE2-7 | Delete project | Project management |
| 16 | RE3-1 | View list of project test suites | Test management |
| 17 | RE3-2 | View test suite detail | Test management |
| 18 | RE3-3 | Create a test suite | Test management |
| 19 | RE3-4 | Edit a test suite | Test management |
| 20 | RE3-5 | Run a test suite | Test management |
| 21 | RE3-6 | Delete a test suite | Test management |
| 22 | RE4-1 | View file | File/Script management |
| 23 | RE4-2 | Upload file | File/Script management |
| 24 | RE4-3 | Delete file | File/Script management |
| 25 | RE4-4 | Download file | File/Script management |
| 26 | RE5-1 | Watch real-time report | Report management |
| 27 | RE5-2 | View final report | Report management |
| 28 | RE5-3 | Export reports | Report management |
| 29 | RE5-4 | View test run history | Report management |
| 30 | RE3-7 | Run a test suite in distributed mode | Test management |
| 31 | RE6-1 | Create new agent | Agent management |
| 32 | RE6-2 | Delete agent | Agent management |
| 33 | RE6-3 | Edit agent | Agent management |
| 34 | RE6-4 | View agent list | Agent management |

The tests would include non-functional requirements tests referred from SRS.

## FEATURES NOT TO BE TESTED

N/A

## TEST OBJECTIVES

Ensuring the functional requirements from the tested feature are met and no defects are found.

The main workflow and critical functions is tested and reported for the stabilization of the product.

Find critical defects and issues on the product and reported.

Crash and system error should be detected and fix.

## TEST ENTRY CRITERIA

* Must have logon credentials & required access to process testing activities.
* TFS account for logging & tracking defects
* Valid Test data is generated by development team to enable testing
* Architecture design & functional Requirement must be made available
* Separate QA environment with its own web server, database and Application server instance must be available
* All standard software tools including testing tools must have been successfully installed and functioning properly

## TEST EXIT CRITERIA

* All tests planned have been run (Manual & Automation)
* No Critical or high severity defects are left outstanding
* High risk areas are completely tested

# Testing levels

## UNIT TESTING

The Unit test should be thorough enough if the product build passes. The Unit test should be cover all the business logical requirements of each units. Developer should immediately modified the code when bugs are founded on test fails until units meet the business requirements and pass the Unit Test cases. Unit Test is done by each member of SCRUM team after the development of module is finished.

## SANITY TESTING

Sanity testing also called as Build Verification Testing (BVT) is a basic test to quickly test whether the build delivered is fit for the further testing. The intention of sanity testing is to rule out certain classes of obviously false results, but not to catch every possible error.

## SYSTEM TESTING

System testing is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s).

During test case design, coverage will be taken care for different aspects like UI, System integration and functional.

* The system testing should be performed across the whole system which will include the individual modules, interfaces and other components.
* This should be a complete end to end testing and the modules/interfaces to be included for system testing and it should again be dependent on the requirement defined by the customer.

Developer tests – System test (deployment, functional) –– Regression.

## AD-HOC/RANDOM TESTING

Testing contrived for only the specific purpose or problem at hand, testing not carefully planned in advance.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Level  (Black box testing) | Planned | Test contents / targets | Done By |
| Developer tests (unit or component tests) | Yes | Test of essential system components, such as individual routines or individual components, including UI components. | Dev Team |
| Sanity Test | Yes | Sanity testing would be performed once the build is deployed in Quality environment in order to evaluate the build and proceed with further testing | Test Team |
| Functional test | Yes | Functional test/assessment of compliance with respect to the specifications. It Would cover individual functionalities described (in detail as per document) within a functional area/module | Test Team |
| Integration/System test | Yes | Integration of individual components and testing the data and communication flow between each components as a whole system | Test Team |
| Regression test | No | Retroactive/repeat test with selected or reduced scope. Critical and major scenarios / test cases are tested to ensure there are no showstoppers. Identification of regression tests along with the test scenario preparation. | Test Team |
| User Acceptance test (Alpha) | Yes | Acceptance of functions or processes in accordance with business requirements specifications/performance specifications, with authorization of the quality environment | Test Team  Dev Team  Operations Team |
| User Acceptance test (Beta) | Yes | Acceptance of functions or processes in accordance with business requirements specifications/performance specifications, with authorization of the productive environment by the user department. | Operations Team / Customer |

## RESOURCES AND BUDGET

All resources required for the tests are included in the overall resource planning. Besides the actual project team, resources that are required only for the test phases (key users, end users etc.) have been included in particular.

All costs for planning and execution of all the test levels are included in the project costs. The costs for usage and implementation of test tools used are also included.

|  |  |  |  |
| --- | --- | --- | --- |
| Planned Date | Details | Responsible | Budget |
| 01/05/2017 to 01/16/2017 | Test Plan | Quan Nguyen (RBVH/ETI1) |  |
| 01/16/2017 to  04/15/2017 | Unit Testing | Quan Nguyen (RBVH/ETI1) |  |
| 04/15/2017 to 05/07/2017 | System testing | Quan Nguyen (RBVH/ETI1) |  |
| 05/07/2017 to 05/15/2017 | Test Case Review /Rework | RBVH/ETI1 - Capstone Team |  |
| 05/15/2017 to 05/25/2017 | User Acceptance Testing | Customer  Capstone Team |  |

|  |  |  |
| --- | --- | --- |
| Role | Resources Recommended | Responsibilities/Comments |
| Tester | 1 | Responsible for testing activities which mentioned in the following :   * Create test plan * Create test cases * Create automated Unit test codes * Execute test * Report the testing progress to Scrum Master * Log Result * Self-monitoring tasks * Analyze and recover from test failures |

# Risk list

List of risk may affect the design, development and implementation of testing

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Mitigation Strategy | Responsible | Contigency ( Risk is realized ) |
| Exit criteria are not met | The tester work with the SCRUM team to define the Exit criteria before starting test. | Tester | Meet outstanding prerequisites  Consider package is not passed the testing |
| Inadequate test data | Ensure a full set of suitable and protected test data is available  Tester will indicate what is required and will verify the suitability of test data | Tester | Redefine test data  Review Test plan and modify |
| Inadequate test cases | The tester will strive to ensure adequate test cases are generated.  Brainstorming with SCRUM team for more test cases | Tester | Redefine test data  Review Test plan and modify  Rewrite test cases and have them reviewed by team. |
| Requirement is not clear. Tester can:  - Not have good estimation because there are many unclear requirements from Client.  - Spend more time to review, update test cases, retest build and release build later than plan because requirement can be changed by Client or BA.  - Loose more time to re-read requirement and prepare some un-useful test cases. | Tester should :  - Raise this risk to Developers  that tester need to clarify more details for unclear requirement as soon as possible to avoid any impacting in preparing test cases, developing application and testing plan.  - Write test cases in general first then go to details later to avoid there are many things changed if your project doesn't have much time for testing. | Developers | - Dev team must update the requirement in more details clearly.  - Tester updates test plan against with the different or new item in understanding requirement, preparing test cases and executing test when the unclear requirement impacts much to current requirement.  - Updates test cases and test plan. |
| Build delivery is late from development team. Tester can:  - Miss and slip test plan due to late build delivery from developer team.  - Not complete tasks on test plan.  - Work OT or reduce testing scope in plan to meet deadline. | Tester should :  - Follow and remind Development leader to prepare build for tester on time as build delivery plan.  - Raise this risk to scrum master to have better solution to satisfy the delivery plan. | Tester | Tester must plan to replace resource in reasonable way.  It must cover plan to study requirement and research site or product. |

# Test case/report templates



# Test environment and implementation

## TEST ENVIRONMENT

|  |  |  |
| --- | --- | --- |
| Item Name | Description | Prepared by |
| Windows 7 – IE11 | Testing environment | RBVH/ETI1 |
| Windows 7 – Firefox 45 | Testing environment | RBVH/ETI1 |
| TestNG | Library | RBVH/ETI1 |
| Junit | Library | RBVH/ETI1 |
| TFS | Bug tracking system | RBVH/ETI1 |

## TEST IMPLEMENTATION AND EXECUTION

Test implementation and execution has the following major tasks:

* Finalizing, implementing and prioritizing test cases
* Verifying and updating bi-directional traceability between the test cases and requirements
* Creating test data
* Writing automated test scripts for efficient test execution
* Verifying that the test environment has been set up correctly
* Executing tests by using automated test scripts
* Comparing actual with expected results & logging the outcome of test execution
* Review test cases with team, report the discrepancies as incidents, and analyze them to find the root cause.

# Suspension criteria and resumption testing

If any defects are found which seriously impact the test progress, the test team may choose to

Suspend testing. Criteria that will justify test suspension are:

* Hardware/software is not available at the times indicated in the project schedule.
* Source code contains one or more critical defects, which seriously prevents or limits testing progress.
* Main functions which are in the release note had not been properly met.
* Resumption will only occur when the problem(s) that caused the suspension has been resolved. When a critical defect is the cause of the suspension, the “FIX” must be verified by the test department before testing is resumed.

If at any point in time issues are submitted that require a design change to the system, all testing will be suspended. After the changes to the requirements, system architecture, and object design are made, a review and updates will be performed of the test specifications to ensure they properly align with the revised system changes.

* After updates are made, testing will resume. Tests in the vicinity of the change must all be rerun. A 20% regression of other tests must also be performed to ensure the changes did not adversely affect other parts of the system

# Test deliverables

The deliverables of the test activities as defined in this Test Plan are outlined in the table below.

|  |  |  |
| --- | --- | --- |
| Deliverable | Owner | Review / Distribution |
| Test Plan | RBVH/ETI1 – Capstone Team | Peer Review |
| Test Cases | RBVH/ETI1 – Capstone Team | Peer Review |
| Test Defect Reports | RBVH/ETI1 – Capstone Team | Peer Review |
| Test Results | RBVH/ETI1 – Capstone Team | Peer Review |

# Test planning, estimates and control

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module | Test Phase | Start Date | End Date | Responsible |
| User Authentication | Test Specification Preparation | 19-Dec-16 | 30-Dec-16 | Capstone Team |
| Test scenarios design | 2-Jan-17 | 13-Jan-17 | Capstone Team |
| Test scripts create | 16-Jan-16 | 27-Jan-17 | Capstone Team |
| Scripts execute | 27-Jan-17 | 30-Jan-17 | Capstone Team |
| Result Analyze | 31-Jan | 3-Feb-17 | Capstone Team |
| Test Report | 6-Feb-17 | 10-Feb-17 | Capstone Team |
| Project Management | Test Specification Preparation | 19-Dec-16 | 30-Dec-16 | Capstone Team |
| Test scenarios design | 2-Jan-17 | 13-Jan-17 | Capstone Team |
| Test scripts create | 16-Jan-16 | 27-Jan-17 | Capstone Team |
| Scripts execute | 27-Jan-17 | 30-Jan-17 | Capstone Team |
| Result Analyze | 31-Jan | 3-Feb-17 | Capstone Team |
| Test Report | 6-Feb-17 | 10-Feb-17 | Capstone Team |
| Test Management | Test Specification Preparation | 19-Dec-16 | 30-Dec-16 | Capstone Team |
| Test scenarios design | 2-Jan-17 | 13-Jan-17 | Capstone Team |
| Test scripts create | 16-Jan-16 | 27-Jan-17 | Capstone Team |
| Scripts execute | 27-Jan-17 | 30-Jan-17 | Capstone Team |
| Result Analyze | 31-Jan | 3-Feb-17 | Capstone Team |
| Test Report | 6-Feb-17 | 10-Feb-17 | Capstone Team |
| File Management | Test Specification Preparation | 19-Dec-16 | 30-Dec-16 | Capstone Team |
| Test scenarios design | 2-Jan-17 | 13-Jan-17 | Capstone Team |
| Test scripts create | 16-Jan-16 | 27-Jan-17 | Capstone Team |
| Scripts execute | 27-Jan-17 | 30-Jan-17 | Capstone Team |
| Result Analyze | 31-Jan-17 | 3-Feb-17 | Capstone Team |
| Test Report | 6-Feb-17 | 10-Feb-17 | Capstone Team |
| Report Management | Test Specification Preparation | 10-Feb-17 | 20-Feb-17 | Capstone Team |
| Test scenarios design | 20-Feb-17 | 3-Mar-17 | Capstone Team |
| Test scripts create | 6-Mar-17 | 17-Mar-17 | Capstone Team |
| Scripts execute | 17-Mar-17 | 20-Mar-17 | Capstone Team |
| Result Analyze | 20-Mar-17 | 23-Mar-17 | Capstone Team |
| Test Report | 23-Mar-17 | 31-Mar-17 | Capstone Team |
| Agent Management | Test Specification Preparation | 3-Apr-17 | 6-Apr-17 | Capstone Team |
| Test scenarios design | 7-Apr-17 | 11-Apr-17 | Capstone Team |
| Test executions | 11-Apr-2017 | 13-Apr-17 | Capstone Team |
| Result Analyze | 14-Apr-2017 | 17-Apr-2017 | Capstone Team |
| Test Report | 18-Apr-2017 | 19-April-17 | Capstone Team |

# Acceptance criteria

Testing activities will be finished after 15-May-2017 milestone

Test cases covered the SRS requirements and critical business logic.

UAT Sign-off from Product Owner for each module

UAT Sign-off from Product Owner for the application