



RHINO

Test Plan

Version: 0.1

**Nov 25 2016**

Author: Khanh Vu

Security Classification:**Confidential**

Software Development & Business Process Services

110 Bishopsgate, London

EC2N 4AY

Tel: +44 (0)20 7333 0033

Email: [info@nashtechglobal.com](mailto:info@nashtechglobal.com)



Table of Contents

[1. Introduction 5](#_Toc476042873)

[1.1 Purpose 5](#_Toc476042874)

[1.2 Glossary 5](#_Toc476042875)

[1.3 Project Context and Background information 6](#_Toc476042876)

[1.4 Document Scope 7](#_Toc476042877)

[1.5 Constraints 7](#_Toc476042878)

[1.6 Risk List 7](#_Toc476042879)

[1.7 Assumptions 11](#_Toc476042880)

[2. Test Approach 12](#_Toc476042881)

[2.1 Test type - Load Test 12](#_Toc476042882)

[2.2 Test Scenarios 12](#_Toc476042883)

[2.3 General Testing Activities 14](#_Toc476042884)

[2.4 Test Schedule 14](#_Toc476042885)

[2.5 Load Testing Process, Status Reporting, Final Report 15](#_Toc476042886)

[2.6 Bug Reporting and Regression Instructions 16](#_Toc476042887)

[3. Resources 16](#_Toc476042888)

[3.1 Human Resources 16](#_Toc476042889)

[3.2 Environment 17](#_Toc476042890)

[4. Deliverables 18](#_Toc476042891)

[5. Suspension / Exit Criteria 18](#_Toc476042892)

[5.1 Suspension Criteria 18](#_Toc476042893)

[5.2 Resumption Criteria 18](#_Toc476042894)

[5.3 Exit Criteria 19](#_Toc476042895)

[6. Appendix 20](#_Toc476042896)

[6.1 Appendix – Performance test 20](#_Toc476042897)

[7. Reference and Related Documents 20](#_Toc476042898)

Document History

| Document Location | |
| --- | --- |
| File Name | Location |
|  |  |

| Document Version History | | | |
| --- | --- | --- | --- |
| Version | Date | Edited by | Details |
| 0.1 | Feb 27th, 2017 |  | Create Test Plan |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Approvers

| Distribution List | | | | |
| --- | --- | --- | --- | --- |
| Name | Role | Signature | Date | Version |
|  |  |  |  |  |
|  |  |  |  |  |

Confidentiality

This document is distributed on a restricted basis, is commercial in confidence to the recipient, and may not be used for any purpose other than that associated with a NashTech project. The contents of this document may not be disclosed to any third parties without the expressed advance written authorisation of NashTech.

# Introduction

[*Note:*

* *Blue or black text within arrow brackets (<< >>) should be customized before publishing this document. Be sure to change the color of the text to black before publishing this document.*
* *Blue text within square brackets ([ ]) provides instructions and guidance and should be deleted before publishing this document.*
* *Blue text is sample. It must be reviewed and customized to meet project requirement*
* *Need to remove all items in “record of change”. It is history of test plan template.* ]

## Purpose

The purpose of this Test Plan is to gather the information necessary to plan and control the test effort for performance testing RHINO website. It describes the approach to testing the software, and will be the top-level plan used by QCs to direct the performance test effort.

This Performance Test Plan for RHINO supports the following objectives:

* Identifies the items that should be targeted by the performance tests and exclusions.
* Scope, resources and schedules of performance testing activities.
* Outlines the performance testing approach that will be used.
* Risk assessment.
* List the deliverable elements of the performance test activities.

## Glossary

| Term | Description |
| --- | --- |
| PM | Project Manager |
| QAO | Quality Assurance Officer |
| QC | Quality Controller |
| UI | User Interface |
| Test Case (TC) | A set of input values, execution preconditions, expected results and execution post conditions, developed for a particular objective or test condition, such as to exercise a particular program path or to verify compliance with a specific requirement |
| Test Objective | Purpose for designing and executing a test or a test type. |
| Test Schedule | A list of activities, tasks or events of the test process, identifying their intended start and finish dates and/or times, and interdependencies. |
| Performance Test | This may include assessing the performance of the test item in terms of transactions per second, throughput per second, response times, round trip time and resource utilization levels.  It serves to investigate measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage. There are many type of performance test such as: performance profiling, load test, stress test, volume test, and other types.  The goal of performance must be defined to specific those items. Otherwise, NashTech considers that performance test is to measure the response time of the system/page with 1 user in specific environment.  See also Performance profiling, Stress testing, Volume Testing, Load Testing |
| Performance Profiling | Performance profiling test is a performance test which subjects the target-of-test to measure the loading time of a web pages or functions of a system, normally this test is executed under one user, the output normally are request sending time, waiting time, response time, render time. |
| Load Testing | A sub-category of performance testing where testing is aimed at assessing the performance of the system when it is placed under specifically increasing loads, usually from low usage conditions, through to typical and peak load conditions e.g. numbers of parallel users and/or numbers of transactions, to determine what load can be handled by the component or system.  See also performance testing, stress Test, Volume Testing |
| Regression Test | Testing of a previously tested program following modification to ensure that defects have not been introduced or uncovered in unchanged areas of the software, as a result of the changes made. It is performed when the software or its environment is changed. |

## Project Context and Background information

Rhino is the software to help owners of small businesses easily manage their work with:

* Clients’ contacts
* Sales proposals
* Projects
* Tasks
* Invoices
* Payment
* Administrative configurations like user management, project types, expense types, system settings, system appearance, etc.

Supported sectors in this application include:

* Construction
* Professional, Scientific & Technical Services:
* Administration & Support Services

## Document Scope

### In Scope

The performance tests will be performed to validate that Rhino system meet the maximum performance standards established for this project. The performance testing is based on the fixed software and defined database. The following functions will be measured for performance:

* Home
* Login
* Add contact
* Add sale proposal
* Add invoice for contact
* Add invoice for Sale proposal
* Add invoice for Empty project
* Add invoice for Project with Task
* Add invoice for Project with Timesheet and Expense
* Add reminder
* Add note
* Add Expense
* Add Timesheet
* Approve Timesheet
* Approve Expense

### Out of Scope

The following requirements are not in scope of testing phase: The performance testing effort outlined in this document will not cover the following:

* Measure the render time on device/browser.
* Any function is out of the Scope.
* Performance testing any changes to Rhino system that are planned.
* Improve performance.

## Constraints

In this project, there are the following constraints:

* Development is delayed. It will affect the progress of the testing.
* More defects are found which require more time to verify and fix.
* Test types which are not mentioned in the test plan will not be tested.
* All test cases will be tested in a popular environment.

## Risk List

List of risks may affect the design, development or implementation of performance testing

| Risk | Mitigation Strategy | Contingency (Risk is realized) |
| --- | --- | --- |
| Exit criteria are not met. | QC(s) works with the PM and QA to define the Exit criteria before start testing. | Meet outstanding prerequisites  Consider package is not passed the testing. |
| Test data proves to be inadequate. | Customer will ensure a full set of suitable and protected test data is available.  QC(s) will indicate what is required and will verify the suitability of test data. | Redefine test data  Review Test Plan and modify |
| More testing packages to test than the plan. | PM agree how many releases for testing | Re-plan  Escalate to the high managements if it impacts to the testing activities. |
| Inadequate test cases. | The QC(s) will strive to ensure adequate test cases are generated. | Redefine test data  Review Test Plan and modify  Rewrite test cases and have them reviewed by the team. |
| The overview of system is large and not easy to understand all | Try to transfer overview system to everybody so that they can understand all processes together | Redefine test data  Review TC for members |
| Requirement is not clear. QC team 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. | QC team should:   * Raise this risk to BA that QC team 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. | * Not have good estimation because there are many unclear requirements from Client. * BA team must update the requirement in more details clearly. * QC team 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. * QC team updates test cases and test plan. |
| Requirement is not completed. QC team can:   * Not have good estimation because there are many uncompleted requirement from Client. * Be out of control the testing scope when requirement is not completed - the same with unclear requirement. * Spend more time to write test cases and release build later than due to adding new function late in requirement -> it leads to quality of build is not good or not meet deadline. | QC team should:   * Raise this risk to BA, PM that QC team need to add more details of uncompleted requirement as soon as possible to avoid impacting writing 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. | * BA team must add more requirements for uncompleted requirement as soon as possible. * QC team updates test plan against with the new item in understanding requirement, preparing test cases and executing test when the uncompleted requirement impacts much to current requirement. * QC team must update test cases and test plan. |
| Performance Test is not clear information or applying NashTech standard. QC team can:   * BA team must add more requirements for uncompleted requirement as soon as possible. * Be not clear about Performance Test (scenarios, expectation, technical infrastructure, server hardware or just note to apply NashTech standard). * Be out of scope the testing effort in preparing Performance Test what needs to be executed, environment, prepared data... | QC team should:   * Raise this risk to BA, PM that QC team need to have more information about Performance Test requirement to scope down of this kind test. | * BA team must discuss with Client and update requirement in more details - match end user expectation and satisfy project scope. * QC team must plan to run Performance Test to generate scenario scripts, prepare environment and test strategy clearly. * QC team re\_plan the Test Plan for Performance Test. |
| Build delivery is late from development team. QC team 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. | QC team should:   * Follow and remind Development leader to prepare build for QC team on time as build delivery plan. * Raise this risk to PM to have better solution to satisfy the delivery plan. | * QC team must re-plan the Test Plan according with real case of late build delivery. * QC team works OT because the plan is slipped to compare with plan before. * Project plan is scoped down due to slipping plan in testing. |
| Communication in team is not good. QC team can:   * Be misunderstanding in requirement, assigned tasks and testing plan. * Take much time to resume information many times for everyone. * Go wrong testing scope. | QC team should:   * Have clear communication in requirement discussion, email, task assignment or chatting to make sure that everyone has the same view in testing activities. * Have meeting together to review or cross check task of each member in team (daily, weekly). | * QC team must update test cases, test plan to cover the misunderstand things. * QC team loses much time to resume the understanding of team many times. |
| Testing Server (Hardware/ software).  Web application can't run well due to difference in installed service versions or other software between Client Server and Testing Server, example: different .Net framework version… | * Project team should have documents for preparing Server environment as Installation Guideline document.   -> This makes sure that there is no any difference between Client Server environment and Test Server environment in hardware / software. | * Project team must have the documents for preparing Server environment as Installation Guideline document. -> This makes sure that there is no difference between Client Server environment and Test Server environment in hardware / software. |

## Assumptions

1. The performance testing will be done on real environment; difference environments will return difference test results.
2. For loading 2,500 simultaneous users, it will be generated by Blazemeter service and the RHINO must be published on the internet.
3. Testing team must have permission to monitor the hardware usage of the production servers while testing.
4. This load testing measure performance of the application while 3,000 simultaneous users accessing to the application, it does not measure the render time on device/browser. As the application is designed with a simple UI so render time should not impact.
5. NashTech we will review the results, diagnose the potential root cause and agree with Eugene on the functions needing to be enhanced before we proceed with any changes.
6. Not any function will be updated while a round of the performance test is running. If this happens, the running round must be stopped and restart again, extra cost will be extended.
7. The performance testing is based on the fixed software and defined database. Additional function added while testing will cause to re-plan and re-schedule.
8. During implementing the automated scripts, if a function could not be applied automation because of technical issue, NashTech will discuss with RHINO to replace another solution if any, otherwise then it will be excluded from the test with informing to RHINO
9. While executing the performance testing, NashTech may have question relates to the business or issue that need confirmation from RHINO, if the answer from RHINO lately affect to the testing progress, then NashTech will re-schedule.
10. RHINO will provide related document such as use case, specification, test case before NashTech start planning. And provide the environment before executing test.
11. Offshore staff will work normal Vietnam working hour.
12. The NashTech process for performance and load test is:

**1. Plan test**

**2. Scripting/ Scenario**

**3. Run test**

**4. Collect statistics**

**5. Analyse result**

**6. Amend system**

RHINO will provide the server’s statistic to NashTech to analyse in order to find root cause of low performance. Improve performance will be out of scope of the testing.

# Test Approach

## Test type - Load Test

A loading test of 2.500 authenticated users (with login function) using the application will be conducted to verify that Rhino website can meet following expectation:

* All response time for user’s click must be less than or equal 3 seconds, except Invoice function it will be 10 seconds because this function take more time to collect data and calculating.

## Test Scenarios

The load testing start with N=100 virtual users then ram up to 2.500 users.

With RHINO system, we need to divide these users to below groups:

|  |  |  |
| --- | --- | --- |
| Groups | Functions | Percentage (%) |
| Normal users | Home | 80% |
| Login |
| Add contact |
| Add sale proposal |
| Add Timesheet |
| Add Expense |
| Add reminder |
| Add note |
| Admin users | Home | 20% |
| Login |
| Approve Timesheet |
| Approve Expense |
| Add invoice for contact |
| Add invoice for Sale proposal |
| Add invoice for Empty project |
| Add invoice for Project with Task |
| Add invoice for Project with Timesheet and Expense |

So, with this classification, we can simulate a reality working where a large number of users accessing to the system.

With this instant, once we increase or reduce the number of simultaneous users for next attempt, the rate will be applied through the testing round.

|  |  |  |
| --- | --- | --- |
| **Step** | **Action** | **Purpose** |
| 1 | Preparation. | Make sure the testing scripts have been implemented. RHINO websites is accessible. Nobody access to the site.  Testing team must have permission to monitor the hardware usage of the test servers while testing.  Testing team must have Blazemeter license. |
| 2 | Generates and load 2.500 users (ram up from 100) send requests to defined functions within 12 hours. | Simulate 2.500 simultaneous users accessing to defined functions to determine if the system still handle the load with the expected performance. |
| 3 | Complete running the script and return a test result | Returns an approximately loading time for all functions which have been run. Or any error message returned while testing running. Analyse and point out where the bottleneck or root causes of the performance issues. |

## General Testing Activities

* Project knowledge training
* Plan the performance test.
* Setup the test environment (Both sides of RHINO and NashTech).
* Implement performance test script
* Execute performance test
* Report test result.

## Test Schedule

High level Test schedule:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Task Name** | **Effort (hours)** | **Resource** | **Start Date** | **End Date** | **Note** | |
| 1 | Training/self-study business requirement and practise | 12 | Senior Tester | 23-Feb-17 | 24-Feb-17 |  |
| 2 | Setup local test environments | 8 | Senior Tester | 24-Feb-17 | 27-Feb-17 | Setup at local testing side, excluding setup test servers which will be done at client side |
| 3 | Create test strategy/test plan document | 12 | Senior Tester | 27-Feb-17 | 28-Feb-17 | Create, review, approve |
| 4 | Test data preparation | 26 | Nam | 6-Mar-17 | 10-Mar-17 | 30 Tenants, 2500/3000 users, 300000 timesheets, 300000 expenses, 3000 projects, 6000 tasks, 3000 Sale proposals, 6000 lines, 3000 contacts. |
| 5 | Create test case | 4.8 | Senior Tester | 1-Mar-17 | 1-Mar-17 | Test happy case only |
| 6 | Implement performance test script | 60 | Senior Tester | 2-Mar-17 | 13-Mar-17 | Support by Tester Leader |
| 7 | Execute Load Test | 15 | Senior Tester | 14-Mar-17 | 15-Mar-17 | 6 rounds with each 2 hours (constraint of tool, does not allow to run 1 round 12 hours), Monitor hardware usage, Collect report between loads. |
| 8 | Analyse, report | 2 | Senior Tester | 16-Mar-17 | 16-Mar-17 | First test report available |
|  | Total (hours) | 147.8 |  |  |  |  |
|  | Total (days) | 18.5 |  |  |  |  |

If hit a bottle neck and need to do enhancements, the recurring effort will be for re-testing is 3.4 days (Testing only, effort to do enhancements from developer team is not included in this estimation).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Task Name** | **Effort (hours)** | **Resource** | **Start Date** | **End Date** | **Note** | |
| 9 | Investigate bottle neck | 8 | Developer & Senior Tester | 17-Mar-17 | 17-Mar-17 | Both developer and tester take part in to this task (each takes 4 hours). |
| 10 | Restore DB and run script to create test data | 2 | Senior Tester | 20-Mar-17 | 20-Mar-17 | Reuse script |
| 11 | Execute Load Test (retest) | 15 | Senior Tester | 20-Mar-17 | 22-Mar-17 | 6 rounds with each 2 hours (constraint of tool, does not allow to run 1 round 12 hours), Monitor hardware usage, Collect report between loads. |
| 12 | Analyse, report | 2 | Senior Tester | 22-Mar-17 | 22-Mar-17 | Final test report available after fixing issues |
|  | Total (hours) | 27 |  |  |  |  |
|  | Total (days) | 3.4 |  |  |  |  |

## Load Testing Process, Status Reporting, Final Report

This section details the load testing process that will be followed for all performance tests conducted as described in this test plan.

The tester will execute all created scripts. These scripts will be generated and executed against the system in 6 rounds with each 2 hours.

We will monitor hardware usage, collect report between loads. We will execute these scripts again, after subsequent hardware, software, or other fixes are introduced.

Test team will baseline load as follows:

We will test Rhino website and report back on the following metrics:

* Response Time each transaction hitting the Web site.
* Any web or database server errors as reported in the data log.
* Failed Web Transactions

There will be Status Reports being sent detailing:

* Performance tests run
* Performance Errors and status
* Number of Bugs Entered
* Status Summary

The Final Report will include summary bug counts, overall performance assessment, and test project summary items.

## Bug Reporting and Regression Instructions

### Bug Reporting:

* Run all load tests and the test result log is generated: view the summary and the details of test result in GUI of Jmeter.
* Analyze the test result: which error occurs and the cause of the error
* Export the test result into JTL, CSV file, PDF or print the report

### Regression:

* Tester does not need to do regression test..

# Resources

## Human Resources

This table shows the staffing assumptions for the project.

| Role | Resource Recommended | Responsibilities / Comments |
| --- | --- | --- |
| QC Lead | 1 | Responsible for all testing activities which mentioned in this document including the following:   * Review the Test plan * Review test cases/ test script prepared by QC. * Support QC in technical issues. * Responsible for QC Technicians training. |
| SQC | 1 | * Create test plan * Create Test cases/Test scripts * Implement performance test script * Execute Load Test * Analyze the results * Investigate bottleneck and diagnose the potential root cause * Prepare Test result report. * Report the testing progress to PM and QCM |
| DEV Team Lead |  | * Test data preparation * Investigate the bottleneck and follow up to make sure that the bug is fixed. |
| Project Management |  | * Provide the documents and monitoring the project progress. * Review/Approve the test plan, test case, test report, final summary report. |
| QC Manager |  | * Handles escalations from senior management and the test team * Approve final summary report. |

## Environment

This section presents the non-human resources required for the Test Plan.

### Testing Environment

[Provide a list of the hardware/software requirement for projects and what will be used in testing.]

3.2.1.1 Hardware / Operating system

Client has not mentioned about the hardware requirement. According to the current plan, the QC team needs to have the following machines.

| Hardware | Operating System | Description |
| --- | --- | --- |
| 1 Test PC | Standard Harvey Nash system | It is using for implementing test script |
| 2 Web servers |  | It is using for hosting the web server |
| 1 DB Server |  | It is using for hosting the database |
|  |  |  |

\*The Servers setup and the load generation will be at RHINO environment. Test server environment separated from production server.

\*Testing team must have permission to monitor the hardware usage of the test servers while testing.

3.2.1.2 Support Tools

Here is list of software which is using in the project.

| Item Name | Description |
| --- | --- |
| Jmeter 3.1 | Supports implement test script. |
| Blazemeter service with license | Supports load generator. Popular license limits 5 load engines, each load engines can generate maximum 500 authenticated users. |

# Deliverables

Here is the list of deliverables in performance testing phase.

| No | Deliverable | Date | Responsibility | Delivered to |
| --- | --- | --- | --- | --- |
| 1 | Test plan | Feb 28th, 2017 | Senior Tester | PM, QCL |
| 2 | Test case | Mar 1st, 2017 | Senior Tester | PM, QCL |
| 3 | Test scripts | Mar 13rd, 2017 | Senior Tester | PM, QCL |
| 4 | First test report | Mar 16th, 2017 | Senior Tester | PM, QCM |
| 5 | Final test report | Mar 22th, 2017 | Senior Tester | PM, QCM |

# Suspension / Exit Criteria

## Suspension Criteria

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.
* Assigned test resources are not available when needed by the test team.
* Main functions which are in the release note had not been properly met.

## Resumption Criteria

If testing is suspended, 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.

## Exit Criteria

Testing will be considered complete when the following conditions have been met:

* All test cases have executed.
* All priority 1 and 2 performance bugs have been resolved and closed.
* No defects must be left in an open unresolved status.

| Severity | Description | Max Number allowed  (depending on system size) | |
| --- | --- | --- | --- |
| For entry into UAT | For exit out of UAT |
| Critical | * Showstopper - An error that makes the whole or a significant part of the application inoperable or otherwise has a significant effect upon Customer's business. * The module/product crashes or the bug causes non-recoverable conditions, database or file corruption, program hangs requiring reboot * Application or service does not work as intended; impossible to use the application or a main function of the application; critical functionality lost * Blocks development and/or testing work (new build, with fix, needed immediately)   e.g. system crash | 0  Showstopper - QC cannot proceed with any tests | 0 |
| Major | * An error that has a material effect upon the functionality, performance or user experience of the application function upon which Customer relies for the efficient conduct of its business. * Major system component unusable due to failure or incorrect functionality, prevents other areas of the app from being * Operation is significantly impacted; User is possibly led to creating future problems. * Can have a work around, but the work around is inconvenient or difficult | 0  QC cannot proceed with any tests in a functional area | 0 |
| Normal | Spelling or grammatical mistakes in the User Interface  Unfriendly behavior that is hindering, but workable, for the user or service  There is a simple work around for the bug | 20  Does not stop testing - but limits it  e.g. data validation wrong | 15 |
| Minor | A minor or cosmetic error that and that causes a minimal effect upon the Customer’s business | 30 | 20 |

# Reference and Related Documents

| No. | Document | Comments | Version |
| --- | --- | --- | --- |
| 1 | PerformanceTest Proposal For Rhino\_v1.0x |  | 1.0 |
| 2 | Rhino Load Test Plan | Option 1 |  |
|  |  |  |  |