System Test Plan

For <<Project Name>>

Votary Softech Solutions Pvt. Ltd.

Plot No: 76, Lumbini layout,   
Near Euro school,   
Gachibowli-I (V), Hyderabad,  
Telangana - 500032,   
India.

**Revision History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version (x.y) | Date of Revision | Description of Change | Reason for Change | Affected Sections | Approved By |
| 1.0 | 22-09-2016 | New Definition | New Process | ALL | Prasanth Vedantam |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Approval History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version (x.y) | Prepared By | Reviewed By/Date | Approved By/Date |
| 1.0 | Rekha Negi | Santosh Palle/ 22-09-2016 | Prasanth Vedantam / 23-09-2016 |
|  |  |  |  |
|  |  |  |  |

**Contents**

[1 Define 4](#_Toc474767832)

[1.1 Overview 4](#_Toc474767833)

[1.2 Test Deliverables 4](#_Toc474767834)

[1.3 Prerequisites 4](#_Toc474767835)

[1.4 Assumptions 4](#_Toc474767836)

[1.5 Limitations 4](#_Toc474767837)

[1.6 Glossary 5](#_Toc474767838)

[2 Specify 5](#_Toc474767839)

[2.1 Test Items 5](#_Toc474767840)

[2.2 Testing Tasks 5](#_Toc474767841)

[2.3 Features to Be Tested 5](#_Toc474767842)

[2.4 Features Not To Tested 5](#_Toc474767843)

[2.5 Team Structure 6](#_Toc474767844)

[2.6 Training Needs 6](#_Toc474767845)

[2.7 Schedule 6](#_Toc474767846)

[2.8 Defect Reporting and Management 6](#_Toc474767847)

[3 Architecture 6](#_Toc474767848)

[3.1 Test Environment Block Diagram 6](#_Toc474767849)

[3.2 Testing Tools 6](#_Toc474767850)

[3.3 Test Configuration Information 6](#_Toc474767851)

[4 Design 7](#_Toc474767852)

[4.1 Approach 7](#_Toc474767853)

[4.2 Entry/Exit Criteria 8](#_Toc474767854)

[4.3 Suspension Criteria and Resumption Requirements 8](#_Toc474767855)

[4.4 Risks and Contingencies 9](#_Toc474767856)

[5 Implement 9](#_Toc474767857)

[6 Validate 9](#_Toc474767858)

[7 Deploy 10](#_Toc474767859)

[8 Maintain 10](#_Toc474767860)

# Define

## Overview

<State the purpose or objective of the test plan for those parties using or referring to the test plan. >

## Test Deliverables

<Define the outputs of the test effort, for example:

* Test Plan
* Test Design Documents
* Test Specifications
* Test Cases/Scripts
* Test Progress Reports
* Defect (Fault) Reports
* Test Results/Summary Reports

Where possible, provide dates for delivery and responsible parties for input to the project schedule. >

|  |  |  |
| --- | --- | --- |
| Item | Date | Responsible |
|  |  |  |
|  |  |  |
|  |  |  |

## Prerequisites

<Prerequisites if any, Simulators, Test data from customer etc.>

## Assumptions

<List all assumptions made with relation to the test effort.>

## Limitations

<List all Limitations to the test effort. Network connectivity, faulty device, data, bandwidth etc. >

## Glossary

<List all Abbreviations and Terms specific to testing >

# Specify

## Test Items

<State the high level list of the product components under test. These will provide input to your testing and form dependencies. Aim to include or reference:

* Name/s of the product/s, system/s and/or parts thereof.
* Software release/version of products.
* Fixes to bugs found in previous releases, or reference to summary Fault logging reports.
* Description of media used to distribute, for example: Web / Server, CD etc.
* Documentation, for example: user, installation, patch guides.>

## Testing Tasks

<List the individual tasks that are needed to prepare for and perform testing. Reference or link to project plans where possible. Load Test, Feature Test, Functional Test>

## Features to Be Tested

<The aim of this section is strategic and is a checklist to provide input for estimating testing effort.

Link each feature to the requirements documents - the source for clarifying client requirements should be the product requirements referenced in the Introduction.

The concept of ‘feature’ encompasses those distinguishing characteristics of the system the client expects. This includes functionality, navigation, usability and forms, as well as aspects of the system such as performance, upgrade and portability, which are not explicitly stated in the requirements documents.)

Aim to base the scope of what is to be tested on highest priority requirements first, for example: test new functionality and code that has been modified to fix or improve old functionality; focus on areas that are most likely to have problems; focus on functions and configurations that will be most used end-users. >

## Features Not To Tested

<The purpose of this section is strategic and states explicitly those items you plan in advance not to cover in testing. Consider:

* Features not covered in this particular release.
* Test configurations you are not going to or are unable to test owing to tight deadlines or business priorities, time and technical constraints.

Note: if unable to test in an operational or exact ‘live’ mode, and will simulate the live environment, state this in the Approach section. Any simulation or restricted configuration will require discussion in the Risk section. >

## Team Structure

<Define the testing team Roles and reporting structure internal to QA and project manager.>

## Training Needs

<Describe the training that you assume will be taken to get the test team the skills needed to perform the planned tests.>

## Schedule

<List the testing milestones, for example: start and end date of testing, date of beta release. Reference can be made to the project plan or standalone schedule, using a link to a URL or path on a network drive. >

## Defect Reporting and Management

<Defects found during testing will be logged in Defect Tracking Tool (Bugzilla/mantis/excel) by Votary Softech QA or Votary Softech developer.

Any problems that are uncovered during system tests will be documented in Votary Softech’s bug tracking system by Votary Softech QA after the <CLIENT> send them to Votary Softech through available communication system.>

# Architecture

## Test Environment Block Diagram

## Testing Tools

<JMeter, loadrunner, Script generator, IDEs if any for Scripting>

## Test Configuration Information

<Describe the hardware and software facilities required to run tests, and provide diagrams where possible to illustrate how the hardware components are connected together. Supply sufficient detail to allow someone to reproduce the test environment and configuration set up. Consider the following when formulating test configuration and environment instances:

* Hardware Configuration
* Software Configuration
* Test Data preparation: All system and unit tests are in the form of explicit actions and expected results. These tests require data matching specific criteria such as a record in a particular state, a field of a required size, or a file with certain data. Finding such test data can involve searching through customer-provided data or creating new data.>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Unit Test Server | Unit Test Clients | System Test Server | System Test Client |
| Hardware |  |  |  |  |
| Processor |  |  |  |  |
| Memory |  |  |  |  |
| Storage |  |  |  |  |
| OS |  |  |  |  |
| Database |  |  |  |  |
| Other Software |  |  |  |  |

# Design

## Approach

<Provide a high level description of how the product is to be tested. Examine each phase of the development life cycle to identify the static and dynamic tests that can be used in the phase.

• Requirements phase – identify all requirements documents that will be produced in this phase, and all inspection results that will be generated as a result of static testing.

• System design phase - identify all requirements documents that will be produced in this phase and plan for test team participation in the associated inspections, along with all inspection results that may be generated as a result of static testing.

• Program design, code, unit test and integration test phases. State the types of white box, memory leak tests, interface tests and cyclamate complexity testing to take place. If the development team is solely responsible for this phase, state their role as an assumption.

• System testing - state the types of testing to be performed: functional, navigation, forms, stress, load, performance, security, volume, installation, upgrade, and backup/restore. State the high level objectives of these tests and how they tie to the requirements or functional specification. If some tests are going to be omitted, state why and relate this to the Risk section.

• Regression testing: describe plans for any regression testing, whether it is on-going as part of maintenance and change requests or part of an incremental release. Describe how regression tests will be chosen and developed, what tools will be used and the strategy for automating the regression testing. Assuming that not all functionality will be covered, describe the scope of regression tests and identify the risks associated with not covering the areas that are omitted.

Also include a statement for phase, on each of the following: agreements with third party suppliers and organizations, together with references to agreements and specific approaches, for example: Risk Based Testing (RBT), bug fix verification methods, use of defect tracking system and reporting method, and any references to test automation - how this test plan fits into a wider test and implementation strategy. >

## Entry/Exit Criteria

<Define the following criteria before the start of system testing:

• Entry Criteria: state clearly what needs to be done before you begin testing. For example: requirements and specification documents finalized, specific test environment and data provided. State whether a smoke test will be performed or provide a checklist of action items that other teams have agreed to provide as an input to testing.

• Exit Criteria: describe what constitutes success of the test phase - what are the conditions that should be met before testing is concluded. For example: ‘All planned tests have been run; all fixed bugs have been verified; all new bugs have been reported; all exceptions to the test plan have been documented.’

• Test Pass/Fail Criteria: state what determines whether a test has passed or not. Specify how you will deal with tests that are not run or are prevented from running.

• Testing Criteria: define testing-specific terms; these might include report statuses, and types and priority of error. >

## Suspension Criteria and Resumption Requirements

<State the intended action should testing be halted by problems - such as severe bugs or environment instability. If possible, state clearly the condition/s under which testing will pause for problems to be fixed and define the statuses to be provide in these circumstances. >

## Risks and Contingencies

<Consider what is most likely to go wrong during the test effort, and then define the plan for fixing the problems that arise. The following are common areas of risk:

* Hardware availability
* Software availability
* Test cases not completed
* Availability of people for testing
* Training or skill levels have not been achieved prior to start of testing
* Requirements changing during development or during testing
* User interface or form style changing during development or during testing

When compiling a list of risks, consider historical data, post-project reviews and even anecdotal evidence. Assess dependencies on other teams to estimate the likelihood of meeting test-entry criteria.

Make reference to Risk Based Testing, where applicable.

Where possible, associate with each risk a probability rating of the risk occurring and a measure of the impact to the project should the risk occur.

Assess the impact of the occurrence of risk to the success of your meeting the test schedule. Use a scale, for example: Critical, Severe, High, Medium or Low, to describe impact.

Develop plans for reducing the probability of occurrence or the impact of risks, beginning with those of a high probability or impact.>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk No | Description | Likelihood/  Severity | Impact | Mitigation |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Implement

<Implementation of Test framework, jar, libraries or utility files using existing IP or New, for teams to use. References to Resource Request for Hardware/Software setup for project, Training. Setup the Environment >

# Validate

<References to Review defect logs, Checklist of Test Plan Document, Approval Emails, Traceability >

# Deploy

<Commits to SVN for team reference

Create project repository for Scripts / Documents and Test environment with approved Plan>

# Maintain

<Changes to Plan with appropriate references>