Test Plan for “Xalts Web App”

*ChangeLog*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Change Date** | **Author** | **Reviewer** | **Approver** | **Description** |
| V1.0 | 3rd Sept 2023 | Arun Vallapu | Arun Vallapu | Abc | Initial Version |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Table of Contents**

Test Plan for “Xalts Web App” 0

1 Introduction 2

1.1 Scope 2

**1.1.1** **In Scope** 2

**1.1.2** **Out of Scope** 2

1.2 Quality Objective 2

1.3 Roles and Responsibilities 2

2 Test Methodology 3

2.1 Overview 3

2.2 Test Levels and types 3

2.3 Defect Management 3

2.4 Entry and Exit Criteria 4

2.5 Suspension Criteria and Resumption Requirements 4

2.6 Test Coverage 4

2.7 Test Automation 4

2.8 Test Execution 5

3 Test Schedule 5

3.1 Milestones 5

4 Test Deliverables 6

5 Resource & Environment Needs 6

5.1 Testing Tools 6

5.2 Test Environment 6

5.3 Team Members 6

6 Assumptions, Risks & Mitigation 7

7 Terms/Acronyms 7

# Introduction

Xalts provides digital assets infrastructure for financial services. Institutions of every size - from large banks to fintechs - use xalts to build their own blockchain based applications.

Open Capital Network (OCN) is built for the users to join the OCN network by onboarding their nodes which can be used as a validator or an RPC for any transactions happening in the network

It also provides users to configure a custom, private and permissioned child network on OCN.

## Scope

### **In Scope**

The initial phase will include all ‘must have’ requirements. Below are the use-cases/scenarios to be tested:

1. Signing Up into the application.
2. Sign In and Sign Out functionalities.
3. Onboarding an OCN Node
4. Launching OCN Child Network
5. Color and Font Family of #3 and #4 scenarios

### **Out of Scope**

Includes features not to be tested that are in development or obsolete.

1. DOCS functionality

## Quality Objective

* Ensure the Application Under Test conforms to functional and non-functional requirements
* Ensure the AUT meets the quality specifications defined by the client
* Bugs/issues are identified and fixed before go live

## Roles and Responsibilities

Detail description of the Roles and responsibilities of different team members like

* QA Analyst – Understand Requirements, prepare Test Plan and Test Cases. Identify scope of automation. Execute testing and manage defects. Implement best practices of software testing.
* Test Manager/Lead - Review test plan, test cases, ensure 100% test coverage, supervise STLC. Provide Test Closure documentation.
* Project Manager – Approve test plan. Monitor testing phases.
* Developers – Build the application as per the requirement specification for Phase1. Ensure the AUT is deployed in the QA environment. Responsible for Unit Testing.
* Infra Team – Provide the test environment and ensure it is up and running. Provide access to project management and defect tracking tools. Install the setup required for test automation (Listed in 4.1)
* Business Analyst/SME – Responsible for defining the functional and non-functional requirements. Responsible for providing clarifications to Dev and QA Teams on specifications.

# Test Methodology

## Overview

The Project delivery methodology defined for OCN is Agile. The testing will be carried out Sprint wise as per the deployment cycle.

## Test Levels and types

Below are the Test Levels that will be performed by the QA Team

1. Integration Testing
2. System Testing
3. User Acceptance Testing

Below are the Testing Types that will be performed by the QA Team

1. Smoke Testing
2. Sanity Testing
3. Regression Testing
4. Usability Testing

*Note : Automated Testing will be performed as part of Smoke and Regression and Usability Testing*

## Defect Management

Below are the guidelines for better defect management:

1. QA team will be logging Bugs for every reproducable test failure
2. QA team will be logging Investigations for inconsistent failures
3. QA team will assign all the tickets to Dev Lead/ Sr. Dev.
4. QA Team will provide the required logs for each ticket.
5. QA Team will provide an update for every changed status of the ticket.
6. Only the QA Team has the authority to close the ticket and no other teams.
7. All the teams will follow the standard [Bug Life Cycle](https://www.softwaretestinghelp.com/bug-life-cycle/).

## Entry and Exit Criteria

QA Team will start the test execution once the below criteria is met

* Test Plan and Test Cases are approved
* Application is up and running in the QA environment

QA Team approves the build for the higher environment and exit testing when the below criteria is met

* All Manual & Automated Test cases executed and passed
* All open QA tickets are fixed or mitigated
* Test closure report is published

## Suspension Criteria and Resumption Requirements

Criteria to be used to suspend all or a portion of the testing activities:

* Unavailability of AUT in the test environment
* A blocker defect that affects 90% of the test cases
* When the response time of the application is more than 2 minutes

Criteria that specify when testing can resume after it has been suspended.

* When AUT environment is restarted and available
* Blocker defects are resolved.
* When the response time is less than 2 minutes

## Test Coverage

Every suite contains positive and negative tests to ensure the build quality and to identify potential issues.

QA Team ensures that 100% of the functional and non-functional requirements are covered in their Regression Suite.

Therefore, it is essential for the team to run the regression cycle to provide test closure.

A requirement traceability matrix will be established by the QA team that points the set of test cases for each Functionality/User Story.

## Test Automation

Below are the guidelines to be followed for Test Automation

* QA Team will identify the scope of automation from the entire test suite
* Test Scenario should be working fine in the manual testing phase
* QA team will paralelly build the test automation for the approved test cases
* QA team will provide the reasons for the test cases that cannot be automated
* Automation framework will be developed by the team and should be approved by the Test Lead
* QA Team will run the automated tests as part of Smoke, Regression and Usability Testing

## Test Execution

**Test Suite**:   
  
QA Team will prepare the master test cases sheet and categorize the test phase and type from that sheet.  
The identified test suite(s) are uploaded to the Test Management tool and are made ready for execution.

**Test cycles**

QA team will execute the test cycles based on the project’s requirements dynamically for each sprint.

**Test Case execution states**

This section defines the state test cases shall take during the test execution phase and what is the meaning of these states. It gives understanding to the stakeholders to comprehend the state in which test cases are.

|  |  |
| --- | --- |
| **Test State** | **Description** |
| Pass | Expected and Actual results of all the test steps are matched |
| Fail | The Expected and Actual results of at least one of the test steps are not matched |
| Blocked | Test case execution is blocked due to a defect or dependency |
| WIP | Test Case execution is in progress |
| Ready | All the test steps are written, and test case is ready to be executed |

# Test Schedule

## Milestones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Name** | **Start** | **Finish** | **Effort (hrs.)** | **Comments** |
| Go through Requirement Documents, Design Documents and get clarifications if any | 4 Sep | 5 Sep | 8 |  |
| Staff and train new test resources | 5 Sep | 11 Sep | 40 | Train on testing concepts |
| First deploy to QA test environment | 11 Sep | 11 Sep | 2 |  |
| Functional testing – Iteration 1 | 11 Sep | 11 Sep | 6 |  |
| Iteration 2 deploy to QA test environment | 12 Sep | 12 Sep | 2 |  |
| Functional testing – Iteration 2 | 12 Sep | 12 Sep | 6 |  |
| System testing | 13 Sep | 15 Sep | 24 |  |
| Regression testing | 18 Sep | 19 Sep | 16 |  |
| UAT | 19 Sep | 19 Sep | 8 |  |

# Test Deliverables

Test Artifacts that will be delivered during different phases of the testing lifecycle.

* Test Plan
* Test Cases
* Requirement Traceability Matrix
* Bug Reports
* Clarification Logs
* Test Strategy
* Test Metrics
* Customer Sign Off

# Resource & Environment Needs

## Testing Tools

* ADO – Project and Test Management
* Selenium Eclipse IDE – Test Automation

## Test Environment

Minimum **hardware** requirements that will be used to test the Application.

* Processor: Intel(R) Core(TM) i5 and above
* RAM: 8GB and above
* System Type: 64-bit operating system, x64-based processor

The following **software is required** in addition to client-specific software.

1. Windows 8 and above
2. MS Office 2013 and above

## Team Members

2.5 FTEs are required as per the below allocation:

|  |  |
| --- | --- |
| 1 FTE | Manual Test Execution and Defect Tracking |
| 1 FTE | Test Automation and Maintenance |
| 0.5 FTE | Documentation |

*Note: Each FTE amounts to 8hrs of productive time.*

# Assumptions, Risks & Mitigation

Assumptions:

* Dev Team has performed Unit testing before deploying the code to QA environment.
* Test resources are finalized.
* Test Schedule are approved by Project Manager
* All the functionalities of Phase1 are available in the QA Environment
* Integration and System testing are only performed on the QA Environment
* UAT is performed in the Pre-Prod environment.

Risks:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Risk** | **Impact** | **Trigger** | **Mitigation Plan** |
| 1 | Testing timeline is limited | High | Might not be able to execute all the test cases | Focus on Targeted testing. Team will prioritize the critical and high-level functionalities |
| 2 | Unplanned leaves | Medium | Delay in test delivery | A backup resource is required for any such instances |

# Terms/Acronyms

| ACRONYM | DEFINITION |
| --- | --- |
| ADO | Azure Dev Ops |
| AUT | Application Under Test |
| OCN | Open Capital Network |
| QA | Quality Assurance |
| UAT | User Acceptance Testing |
| WIP | Work In Progress |
| FTE | Full Time Equivalent |