|  |
| --- |
| Synechron logo - Whitebase |
| Agile Test Plan |
|  |
|  |
| **Synechron** |
|  |

|  |
| --- |
|  |

**Table of Contents**

[1. Introduction 4](#_Toc352756852)

[1.1 Purpose of this document 4](#_Toc352756853)

[2. Acronyms 5](#_Toc352756854)

[3. Test Approach 6](#_Toc352756855)

[3.1 Feature to be tested in different Sprints 6](#_Toc352756856)

[3.2 Identification of impediments 6](#_Toc352756857)

[3.3 Retrospective meetings 6](#_Toc352756858)

[4. Scope of Testing 7](#_Toc352756859)

[4.1 Features to be tested 7](#_Toc352756860)

[4.2 Features not to be tested 8](#_Toc352756861)

[5. Entry and Exit Criteria 9](#_Toc352756862)

[6. Test Environment 10](#_Toc352756863)

[6.1 Hardware Requirements 10](#_Toc352756864)

[6.2 Software Requirements 10](#_Toc352756865)

[7. Tools 12](#_Toc352756866)

[8. Resource Management 13](#_Toc352756867)

[8.1 Roles and Responsibilities 13](#_Toc352756868)

[8.2 Sprint Schedule 13](#_Toc352756869)

[9. Assumptions 14](#_Toc352756870)

[10. Risks & Issues 15](#_Toc352756871)

[11. Test Deliverables 16](#_Toc352756872)

[12. Appendix 17](#_Toc352756873)

[12.1 Defect Classification 17](#_Toc352756874)

Version

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Description |
| November 23, 2012 | 0.1 | QA CoE | Initial Version |
| November 24, 2012 | 1.0 | QA CoE | Final Version after review |

Reviewed By

|  |  |
| --- | --- |
| Date | Name |
| November 24, 2012 | Venu M.S |

# Introduction

The Test Plan has been created to communicate the test approach to team members. It includes the objectives, scope, schedule, risks and approach. This document will clearly identify what the test deliverables will be and what is deemed in and out of scope.

## Purpose of this document

The Software Test Plan (STP) covers the following points:

* + Identify all the activities involved in testing
  + Training of resources required for executing test activities and monitoring mechanisms
  + The strategy and approach of testing the software components along with risk and mitigation plan
  + The defect tracking system to be used with the help of various tools

# Acronyms

This section provides the description of acronym that is required to interpret the STP properly.

| S. No. | Acronym | Description |
| --- | --- | --- |
|  | STP | Software Test Plan |
|  | CCB | Change Control Board |
|  | QA | Quality Assurance |

<<Add all acronyms to be used in the document. >>

# Test Approach

The project is using an agile approach, with weekly iterations. At the end of each week the requirements identified for that iteration will be delivered to the team and will be tested.

## Feature to be tested in different Sprints

<< Identification of story points to be used to each sprint.

Status of sprint back log >>

## Identification of impediments

<< What impediments have been faced in each sprint and how did we overcome it. >>

## Retrospective meetings

<<We discuss the lessons learnt. >>

# Scope of Testing

## Features to be tested

**Module 1:** <<Module Name>>

| S. No. | Features to be tested |
| --- | --- |
|  |  |
|  |  |
|  |  |

**Module 2:** <<Module Name>>

| S. No. | Features to be tested |
| --- | --- |
|  |  |
|  |  |

**Module 3:** <<Module Name>>

| S. No. | Features to be tested |
| --- | --- |
| 1. |  |
| 2. |  |

## Features not to be tested

**Module 1:** <<Module Name>>

| S. No. | Features not to be tested |
| --- | --- |
| 1. |  |

**Module 2:** <<Module Name>>

| S. No. | Features not to be tested |
| --- | --- |
| 1. |  |

**Miscellaneous:**

| S. No. | Features not to be tested |
| --- | --- |
| 1. |  |

# Entry and Exit Criteria

<<Entry and Exit criteria should be defined here

Entry Criteria Example:

1. Test case preparation to start after business requirement sign off.
2. Release notes to be published by development team before test execution.

Exit Criteria Example:

1. All high priority and severity defects are closed.
2. All Test cases are executed at least once.

>>

# Test Environment

## Hardware Requirements

| S. No. | Purpose | Configuration | Quantity |
| --- | --- | --- | --- |
|  | Client | IBM Compatible PC with at least 10 GB HDD, 512 MB RAM | 1 |
|  | Server | IBM Compatible PC with at least 4 GB HDD, 256 MB RAM | 1 |

## Software Requirements

| S. No. | Name of the Software | Version/Release details | No. Of licenses |
| --- | --- | --- | --- |
| **Server Side** | | | |
|  | SQL Server with SQL Server Reporting Service | 2000, with latest service pack | NA |
|  | Windows OS | 2000 professional, Latest SP | 1 |
|  | MS. NET framework | 1.1 | NA |
|  | MQ Series client |  |  |
| **Client Side** | | | |
| 1. | Windows OS | XP | 1 |
| 2. | MS .NET framework without Visual Studio | 1.1 | NA |
| 3. | MS Excel | 2000/XP | 1 |
| 4. | MS Access | 97 | 1 |

# Tools

<<This section will provide details about the various tools (Test Management Tools, Automation Tools, Defect Tracking Tool, etc.) used for Testing.>>

|  |  |  |
| --- | --- | --- |
| S. No. | Name of tool | Usage |
| 1. |  |  |
| 2. |  |  |

# Resource Management

## Roles and Responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| S. No. | Roles | Name of the Resource | Responsibilities |
|  | Business Analyst |  | Responsible for requirement gathering and related query resolution. |
|  | Product Manager |  | Serve as a primary contact/liaison between the development team and the project test team. |
|  | Scrum Master |  | -Responsible for all activities  -Coordinate weekly meetings and will communicate the testing status to the project team  -Create and update the STP |
|  | Testers |  | Responsible for writing, executing test cases and reporting defects. |
|  | Developers |  | Develop the application and perform unit testing. |

## Sprint Schedule

<<Mention the planned Sprint Schedule here>>

# Assumptions

<<Mention project Assumptions and Dependencies here

Example:

1. Business Requirements are signed off before Test Case preparation start date
2. Test Environment should be ready and available prior to test execution activity.>>

# Risks & Issues

<<Mention project Risks and Issues here

Example:

1. Hardware not yet procured

2. Unavailability of domain specific resources >>

|  |  |  |  |
| --- | --- | --- | --- |
| Risk & Issue | Probability | Impact | Mitigation |
|  |  |  |  |
|  |  |  |  |

# Test Deliverables

|  |  |  |
| --- | --- | --- |
| Deliverable | For | Date / Milestone |
| Test Plan | Product Manager; Scrum Master; Team |  |
| Traceability Matrix | Product Manager; Scrum Master; Team |  |
| Test Results | Scrum Master |  |
| Test Status report | Scrum Master |  |
| Metrics | All team members |  |

# Appendix

## Defect Classification

Following are the Defect Priority levels:

1. **High:** The defect must be resolved as soon as possible preferably before next build because it is impairing development and / or testing activities. System use will be severely affected until the defect is fixed.
2. **Medium:** The defect should be resolved in the normal course of development activities. It can wait until a new build or version is created.
3. **Low:** The defect repair can be put of indefinitely. It can be resolved in a future major system revision or not resolved at all.

Following are the Defect Severity levels:

1. **Show Stopper:** The defect results in crash or shut down of the application. The user cannot use the application until the defect is resolved.
2. **Critical:** The defect results in the failure of the complete software system, of a subsystem, or of a software unit (program or module) within the system. There is no workaround.
3. **Major:** The defect does not result in a failure, but causes the system to produce incorrect, incomplete, or inconsistent results, or the defect impairs the systems usability. There is a workaround.
4. **Cosmetic:** The defect is the result of non-conformance to a standard, is related to the aesthetics of the system, or is a request for an enhancement. Mostly user interface issues. Defects at this level may be deferred.