



# Technical Architecture Document

**Customer Onboarding Application**

Version No. 1

## Revision History

Date	Version#	Author	Comments
25/06/2018	1.0.0	Abhilash	Draft

## Document Approval

Signature	Printed Name	Title	Date

# CONTENTS

<b>1. INTRODUCTION.....</b>	<b>1</b>
1.1. Purpose.....	1
1.2. Project Overview.....	1
1.3. Audience.....	2
<b>2. TEST STRATEGY.....</b>	<b>2</b>
2.1. Test Objectives.....	2
2.2. Test Assumptions.....	2
2.3. Test Principles.....	3
2.4. Data Approach.....	4
2.5. Scope and Levels of Testing.....	4
<b>3. TEST PLAN.....</b>	<b>6</b>
<b>4. TEST ARCHITECTURAL REPRESENTATION.....</b>	<b>8</b>
<b>5. AUTOMATION ARCHITECTURE WITH TOOLS.....</b>	<b>9</b>
<b>6. EXECUTION STRATEGY.....</b>	<b>10</b>
6.1. Entry and Exit Criteria.....	10
6.2. Test Cycles.....	11
6.3. Validation and Defect Management.....	11
6.4. Test Metrics.....	12
6.5. Defect tracking & Reporting.....	13
<b>7. TEST MANAGEMENT PROCESS.....</b>	<b>13</b>
7.1. Test Management Tool.....	13
7.2. Test Design Process.....	14
7.3. Test Execution Process.....	15
<b>8. GITHUB AUTOMATION SOURCE CODE REPOSITORY.....</b>	<b>17</b>

## 1. INTRODUCTION

### 1.1. Purpose

This test plan describes the testing approach and overall framework that will drive the testing of the **Customer OnBoarding Application** Version 1.0 – My Info "<https://customeronboardapp.zendesk.com>" site. The document introduces:

This document provides a high level overview and explains the whole architecture of **Customer OnBoarding**. It explains how an online user will be able to create new users and migrate the customers to this platform. How to customize the views. How to API calls to the application etc. The document provides a high-level description of the architecture, the use cases support by the system and architectural styles and components that have been selected to best achieve the use cases. This framework then allows for the development of the design criteria and documents that define the technical and domain standards in detail.

- **Login and Views**
- **Import Users**
- **API Calls**
  
- Test Strategy: rules the test will be based on, including the givens of the project
  - Start Date : June 27, 2018
  - End Date : July 24, 2018
  - No. Of Resources : 4
- Execution Strategy: describes how the test will be performed and process to identify and report defects, and to fix and implement fixes.
- Test Management: process to handle the logistics of the test and all the events that come up during execution (e.g.: communications, escalation procedures, risk and mitigation, team roster)

### 1.2. Project Overview

Customer On-boarding Module is a powerful tool providing the user to carry out various procurement's. It allows the tenants to import the existing user with the help of an Excel Template while migrating to this platform.

The functionality of this module spans through the entire system, making information available anywhere, anytime. An User can customize the views and the customized details are stored in the database.

### 1.3. Audience

- Project team members perform tasks specified in this document, and provide input and recommendations on this document.
- Project Manager Plans for the testing activities in the overall project schedule, reviews the document, tracks the performance of the test according to the task herein specified, approves the document and is accountable for the results.
- The stakeholders' representatives and participants may take part in the UAT test to ensure the business is aligned with the results of the test.
- Technical Team ensures that the test plan and deliverables are in line with the design, provides the environment for testing and follows the procedures related to the fixes of defects.
- Business analysts will provide their inputs on functional changes.

## 2. TEST STRATEGY

### 2.1. Test Objectives

The objective of the test is to verify that the functionality of Customer On-Boarding 1.0 works according to the specifications.

The test will execute and verify the test scripts, identify, fix and retest all high and medium severity defects per the entrance criteria, prioritize lower severity defects for future fixing via CR.

The final product of the test is twofold:

- A production-ready software;
- A set of stable test scripts that can be reused for Functional and UAT test execution.

### 2.2. Test Assumptions

#### Key Assumptions

- Production like data required and be available in the system prior to start of Functional Testing
- In each testing phase, Cycle 3 will be initiated if the defect rate is high in Cycle 2.

#### General

- Exploratory Testing would be carried out once the build is ready for testing
- Performance testing is not considered for this estimation.
- All the defects would come along with a snapshot JPEG format
- The Test Team will be provided with access to Test environment
- The Test Team assumes all necessary inputs required during Test design and execution will be supported by Development/BUSINESS ANALYSTs appropriately.

- Test case design activities will be performed by QA Group
  - Test environment and preparation activities will be owned by Dev Team
  - Dev team will provide Defect fix plans based on the Defect meetings during each cycle to plan. The same will be informed to Test team prior to start of Defect fix cycles
  - BUSINESS ANALYST will review and sign-off all Test cases prepared by Test Team prior to start of Test execution
  - The defects will be tracked through HP ALM only. Any defect fixes planned will be shared with Test Team prior to applying the fixes on the Test environment
  - Project Manager/BUSINESS ANALYST will review and sign-off all test deliverables
  - The project will provide test planning, test design and test execution support
  - Test team will manage the testing effort with close coordination with Project PM/BUSINESS ANALYST
  - Project team has the knowledge and experience necessary, or has received adequate training in the system, the project and the testing processes.
  - There is no environment downtime during test due to outages or defect fixes.
- 
- The system will be treated as a black box; if the information shows correctly online and in the reports, it will be assumed that the database is working properly.
  - Cycle 3 will be initiated if there are more defects in Cycle 2.

### **Functional Testing**

- During Functional testing, testing team will use preloaded data which is available on the system at the time of execution
- The Test Team will be perform Functional testing only on Customer On-Boarding 1.0 MODULE

### **UAT**

- UAT test execution will be performed by end users (L1, L2 and L3) and QA Group will provide their support on creating UAT script.

### **Modules to be Tested**

The purpose of test plan document is to provide details on how testing process will be conducted for the following modules.

- **Login and Views**
- **Import Users**
- **API Calls**

## **2.3. Test Principles**

- Testing will be focused on meeting the business objectives, cost efficiency, and quality.
- There will be common, consistent procedures for all teams supporting testing activities.
- Testing processes will be well defined, yet flexible, with the ability to change as needed.
- Testing activities will build upon previous stages to avoid redundancy or duplication of effort.

- Testing environment and data will emulate a production environment as much as possible.
- Testing will be a repeatable, quantifiable, and measurable activity.
- Testing will be divided into distinct phases, each with clearly defined objectives and goals.
- There will be entrance and exit criteria.

## 2.4. Data Approach

- In functional testing, Customer On-Boarding 1.0 MODULE will contain pre-loaded test data and which is used for testing activities.

## 2.5. Scope and Levels of Testing

### 2.5.1. Test Coverage

- Functional Testing
- Regression Testing (Selenium WebDriver & TestNG)
- Performance Testing (JUnit & JMeter)
- Endurance Testing (as per the requirement)
- API Testing (PostMan)
- Security Testing (as per the requirement)






### 2.5.2. Performance

The system should be able to support 1000 tenants uploading 50000 customers simultaneously without any application crash or performance degrade. Example content: [There is no particular constraints related to system performance. It is anticipated that the system should respond to any request well under standard database and web server script timeouts (20 seconds), also system performance can depend on available hardware, network and internet connection capabilities. In addition, upload / download times can depend on data size which in turn depends on user input. Therefore, actual performance can be determined only after system deployment and testing.

Performance Test Suite can be developed in JUnit and can simulate the simultaneous import of 50000 customers by 1000 tenants using JMeter. Also use PerfMon to monitor the Memory Usage to observe any Memory Leak issues.

### 2.5.3. Use Case View

This is a list of use-cases that represent major functionality of the final system [SRS]:

-  User Login
-  View process specification
-  Migrating users
-  Application performance with simultaneous migration
-  API Calls

## Automation Framework

- Hybrid Framework (TestNG, JUnit, Data driven)

## Number of Resources and the required skill sets

- Test Engineer (Two resources for both Manual and Automation with the core skill sets mentioned under the Automation Architecture Tools)

### TEST ACCEPTANCE CRITERIA

- Approved Functional Specification document, Use case documents must be available prior to start of Test design phase.
- Test cases approved and signed-off prior to start of Test execution
- Development completed, unit tested with pass status and results shared to Testing team to avoid duplicate defects
- Test environment with application installed, configured and ready to use state

Sign-off	Readiness
<ul style="list-style-type: none"> <li>•Approved Functional Specification Document</li> <li>•Approved Use cases</li> <li>•Approved Test cases</li> </ul>	<ul style="list-style-type: none"> <li>•Development completed &amp; unit tested</li> <li>•Application deployed and system ready for testing on Test environment</li> <li>•Production like data is available to test all functionalities.</li> <li>•Defect fixes planned based on Defect triage (Unit Testing) and evaluation criteria</li> </ul>

S.No.	Deliverable Name	Author	Reviewer
1.	Test Plan	Test Lead	Project Manager/ Business Analyst's
2.	Functional Test Cases	Test Team	Business Analyst's Sign off
3.	Logging Defects in HP ALM	Test Team	Test Lead/ Programming Lead(Vijay)
(4.	Daily/weekly status report	Test Team/ Test Lead	Test Lead/ Project Manager
5.	Test Closure report	Test Lead	Project Manager

### MILESTONE LIST

The milestone list is tentative and may change due to below reasons

- Any issues in the System environment readiness
- Any change in scope/addition in scope
- Any other dependency that impacts efforts and timelines



#### 2.5.4. User Acceptance Test (UAT)

**PURPOSE:** this test focuses on validating the business logic. It allows the end users to complete one final review of the system prior to deployment.

**TESTERS:** the UAT is performed by the end users (L1, L2 and L3).

**METHOD:** Since the business users are the most indicated to provide input around business needs and how the system adapts to them, it may happen that the users do some validation not contained in the scripts. Test team write the UAT test cases based on the inputs from End user (L1,L2 and L3 users) and Business Analyst's.

**TIMING:** After all other levels of testing (Exploratory and Functional) are done. Only after this test is completed the product can be released to production.

#### **TEST DELIVERABLES**

S.No.	Deliverable Name	Author	Reviewer
1.	UAT Test Cases	Test Team	Business Analyst's Sign off

### 3. TEST PLAN

#### 3.1. Test Coverage

- Functional Testing
- Regression Testing (Selenium WebDriver & TestNG)
- Performance Testing (JUnit & JMeter)
- Endurance Testing (as per the requirement)
- API Testing (PostMan)
- Security Testing (as per the requirement)

#### 3.2. Automation Framework

- Hybrid Framework (TestNG, JUnit, Data driven)

### 3.3. Number of Resources and the required skill sets

- **Test Engineer (Two resources for both Manual and Automation with the core skill sets mentioned under the Automation Architecture Tools)**

### 3.4. Requirement Traceability Matrix



RTM\_OnBoardApp.xlsx

#### Test Plan

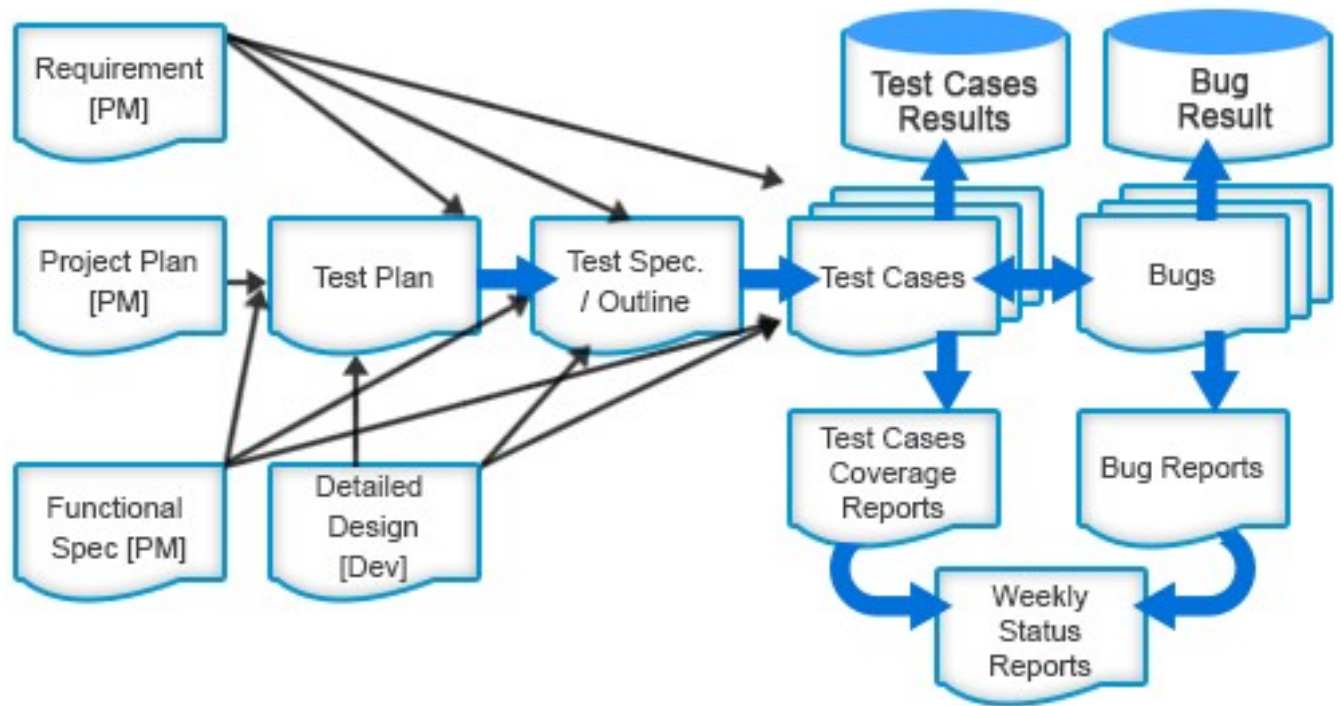


TC\_OnBoardApp.xlsx

### 3.5. Handling dynamic screens

- **Element with Index** - Each web element are located in different hierarchical level and we can search specific element from similar elements within hierarchical by using level index.
- **Absolute XPath Method**
- **Relative XPath Method**
- **Relative XPath either with either Multiple Attributes, with Starting Text, with contains Text etc.**

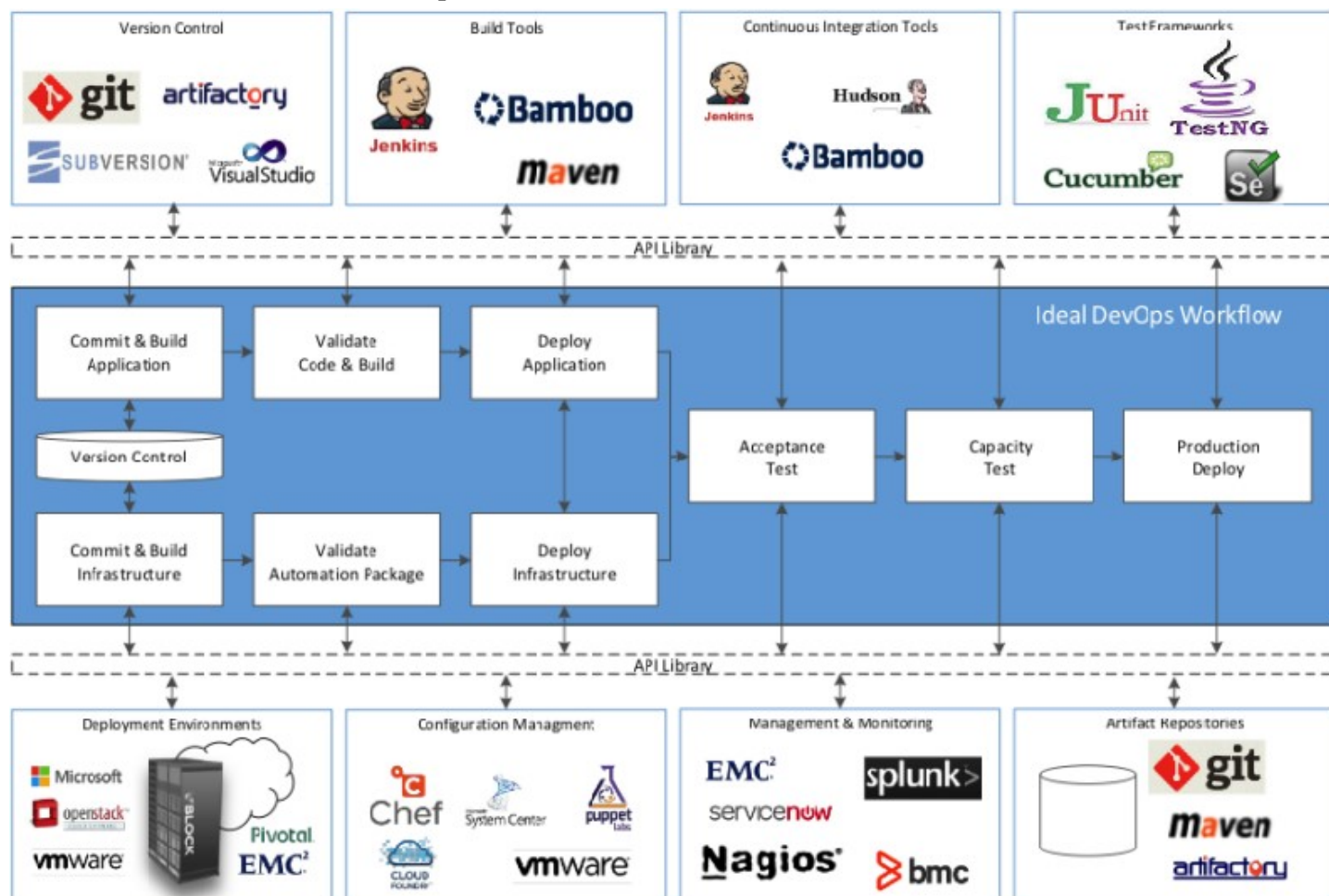
#### 4. TEST ARCHITECTURAL REPRESENTATION



## 5. AUTOMATION ARCHITECTURE WITH TOOLS

- **Selenium WebDriver with Java** - Functional Testing
- **JMeter** - Performance Testing
- **PostMan** - Rest API Testing
- **TestNG** - Test Automation Framework
- **JUnit + JMeter + PerfMon** - Test Framework for Performance Testing
- **GitHub** - Source Code Repository
- **Maven** - Build Tool
- **Rational ClearQuest\BugZilla, JIRA** - Bug Tracking
- **HP ALM\JIRA** - Test Management
- **Jenkins** - Deployment Tool (Can be configured for unattended nightly build creation, deployment and Regression Test Execution)

### Automation Workflow with option for other tools



## 6. EXECUTION STRATEGY

### 6.1. Entry and Exit Criteria

- The entry criteria refer to the desirable conditions in order to start test execution; only the migration of the code and fixes need to be assessed at the end of each cycle.
- The exit criteria are the desirable conditions that need to be met in order proceed with the implementation.
- Entry and exit criteria are flexible benchmarks. If they are not met, the test team will assess the risk, identify mitigation actions and provide a recommendation. All this is input to the project manager for a final “go-no go” decision.
- Entry criteria to start the execution phase of the test: the activities listed in the Test Planning section of the schedule are 100% completed.
- Entry criteria to start each cycle: the activities listed in the Test Execution section of the schedule are 100% completed at each cycle.

Exit Criteria	Test Team	TechnicalTeam	Notes
100% Test Scripts executed			
95% pass rate of Test Scripts			
No open Critical and High severity defects			
95% of Medium severity defects have been closed			
All remaining defects are either cancelled or documented as Change Requests for a future release			
All expected and actual results are captured and documented with the test script			
All test metrics collected based on reports from HP ALM			
All defects logged in HP ALM			
Test Closure Memo completed and signed off			
Test environment cleanup completed and a new back up of the environment			

## 6.2. Test Cycles

- There will be two cycles for functional testing. Each cycle will execute all the scripts .
- The objective of the first cycle is to identify any blocking, critical defects, and most of the high defects. It is expected to use some work-around in order to get to all the scripts.
- The objective of the second cycle is to identify remaining high and medium defects, remove the work-around from the first cycle, correct gaps in the scripts and obtain performance results.
- π UAT test will consist of one cycle.

## 6.3. Validation and Defect Management

- It is expected that the testers execute all the scripts in each of the cycles described above. However it is recognized that the testers could also do additional testing if they identify a possible gap in the scripts. This is especially relevant in the second cycle, when the Business analyst's join the TCOE in the execution of the test, since the BUSINESS ANALYSTs have a deeper knowledge of the business processes. If a gap is identified, the scripts and traceability matrix will be updated and then a defect logged against the scripts.
- The defects will be tracked through HP ALM only. The technical team will gather information on a daily basis from HP ALM, and request additional details from the Defect Coordinator. The technical team will work on fixes.
- It is the responsibility of the tester to open the defects, link them to the corresponding script, assign an initial severity and status, retest and close the defect; it is the responsibility of the Defect Manager to review the severity of the defects and facilitate with the technical team the fix and its implementation, communicate with testers when the test can continue or should be halt, request the tester to retest, and modify status as the defect progresses through the cycle; it is the responsibility of the technical team to review HP ALM on a daily basis, ask for details if necessary, fix the defect, communicate to the Defect Manager the fix is done, implement the solution per the Defect Manager request.

Defects found during the Testing will be categorized according to the bug-reporting tool "Mercury HP ALM" and the categories are:

Severity	Impact
1 (Critical)	<ul style="list-style-type: none"> <li>This bug is critical enough to crash the system, cause file corruption, or cause potential data loss</li> <li>It causes an abnormal return to the operating system (crash or a system failure message appears).</li> <li>It causes the application to hang and requires re-booting the system.</li> </ul>
2 (High)	<ul style="list-style-type: none"> <li>It causes a lack of vital program functionality with workaround.</li> </ul>
3 (Medium)	<ul style="list-style-type: none"> <li>This Bug will degrade the quality of the System. However there is an intelligent workaround for achieving the desired functionality - for example through another screen.</li> <li>This bug prevents other areas of the product from being tested. However other areas can be independently tested.</li> </ul>
4 (Low)	<ul style="list-style-type: none"> <li>There is an insufficient or unclear error message, which has minimum impact on product use.</li> </ul>
5(Cosmetic)	<ul style="list-style-type: none"> <li>There is an insufficient or unclear error message that has no impact on product use.</li> </ul>

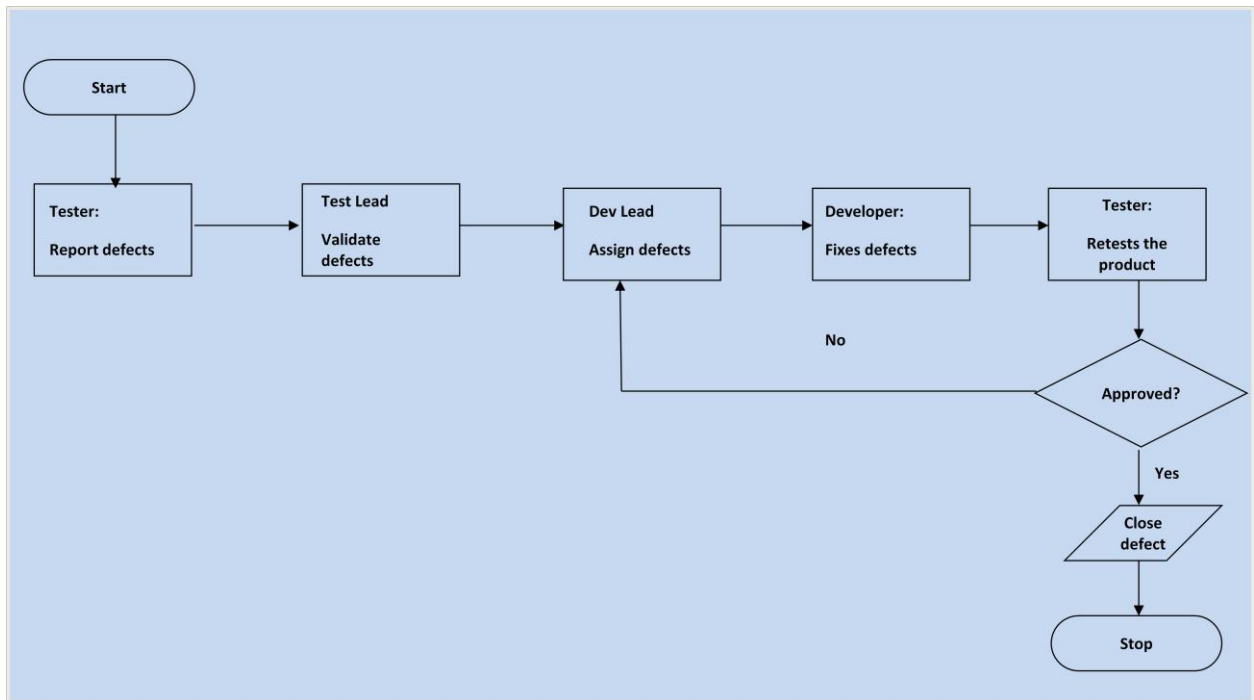
#### 6.4. Test Metrics

Test metrics to measure the progress and level of success of the test will be developed and shared with the project manager for approval. The below are some of the metrics

Report	Description	Frequency
Test preparation & Execution Status	To report on % complete, %WIP, % Pass, % Fail Defects severity wise Status – Open, closed, any other Status	Weekly / Daily (optional)
Daily execution status	To report on Pass, Fail, Total defects, highlight Showstopper/ Critical defects	Daily
Project Weekly Status report	Project driven reporting (As requested by PM)	Weekly – If project team needs weekly update apart from daily and there is template available

## 6.5. Defect tracking & Reporting

Following flowchart depicts Defect Tracking Process:



## 7. TEST MANAGEMENT PROCESS

### 7.1. Test Management Tool

HP Application Lifecycle Management is the tool used for Test Management. All testing artifacts such as Test cases, test results are updated in the HP Application Lifecycle Management (ALM) tool.

- Project specific folder structure will be created in HP ALM to manage the status of this DFRT project.
- Each resource in the Testing team will be provided with Read/Write access to add/modify Test cases in HP ALM.
- During the Test Design phase, all test cases are written directly into HP ALM. Any change to the test case will be directly updated in the HP ALM.
- Each Tester will directly access their respective assigned test cases and update the status of each executed step in HP ALM directly.
- Any defect encountered will be raised in HP ALM linking to the particular Test case/test step.



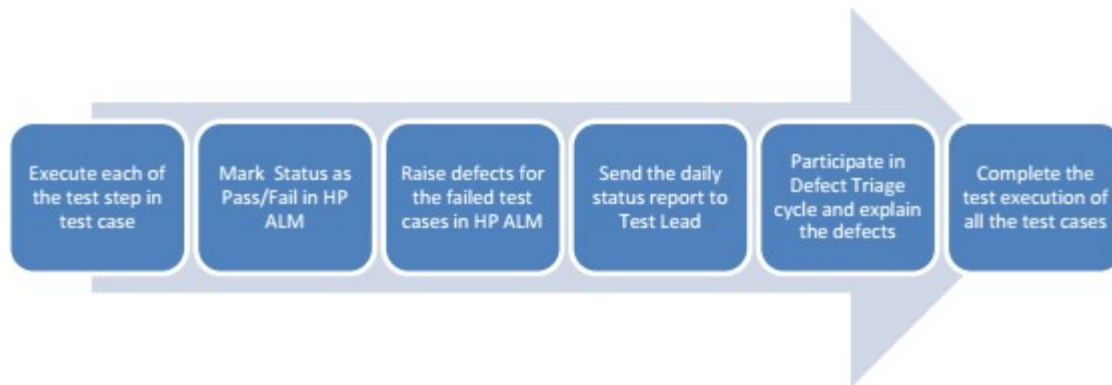
- During Defect fix testing, defects are re-assigned back to the tester to verify the defect fix. The tester verifies the defect fix and updates the status directly in HP ALM.
- Various reports can be generated from HP ALM to provide status of Test execution. For example, Status report of Test cases executed, Passed, Failed, No. of open defects, Severity wise defects etc.

## 7.2. Test Design Process



- The tester will understand each requirement and prepare corresponding test case to ensure all requirements are covered.
- Each Test case will be mapped to Use cases to Requirements as part of Traceability matrix.
- Each of the Test cases will undergo review by the BUSINESS ANALYST and the review defects are captured and shared to the Test team. The testers will rework on the review defects and finally obtain approval and sign-off.
- During the preparation phase, tester will use the prototype, use case and functional specification to write step by step test cases.
- Testers will maintain a clarification Tracker sheet and same will be shared periodically with the Requirements team and accordingly the test case will be updated. The clarifications may sometimes lead to Change Requests or not in scope or detailing implicit requirements.
- Sign-off for the test cases would be communicates through mail by Business Analyst's.
- Any subsequent changes to the test case if any will be directly updated in HP ALM.

### 7.3. Test Execution Process



- Once all Test cases are approved and the test environment is ready for testing, tester will start a exploratory test of the application to ensure the application is stable for testing.
- Each Tester is assigned Test cases directly in HP ALM.
- Testers to ensure necessary access to the testing environment, HP ALM for updating test status and raise defects. If any issues, will be escalated to the Test Lead and in turn to the Project Manager as escalation.
- If any showstopper during exploratory testing will be escalated to the respective development Team for fixes.
- Each tester performs step by step execution and updates the executions status. The tester enters Pass or Fail Status for each of the step directly in HP ALM.
- Tester will prepare a Run chart with day-wise execution details
- If any failures, defect will be raised as per severity guidelines in HP ALM tool detailing steps to simulate along with screenshots if appropriate.
- Daily Test execution status as well as Defect status will be reported to all stakeholders.
- Testing team will participate in defect triage meetings in order to ensure all test cases are executed with either pass/fail category.
- If there are any defects that are not part of steps but could be outside the test steps, such defects need to be captured in HP ALM and map it against the test case level or at the specific step that issue was encountered after confirming with Test Lead.
- This process is repeated until all test cases are executed fully with Pass/Fail status.
- During the subsequent cycle, any defects fixed applied will be tested and results will be updated in HP ALM during the cycle.

As per Process, final sign-off or project completion process will be followed

#### 7.4. Test Execution Process

Risk	Prob.	Impact	Mitigation Plan
<b>SCHEDULE</b> Testing schedule is tight. If the start of the testing is delayed due to design tasks, the test cannot be extended beyond the UAT scheduled start date.	High	High	<ul style="list-style-type: none"> <li>The testing team can control the preparation tasks (in advance) and the early communication with involved parties.</li> <li>Some buffer has been added to the schedule for contingencies, although not as much as best practices advise.</li> </ul>
<b>RESOURCES</b> Not enough resources, resources on boarding too late (process takes around 15 days).	Medium	High	Holidays and vacation have been estimated and built into the schedule; deviations from the estimation could derive in delays in the testing.
<b>DEFECTS</b> Defects are found at a late stage of the cycle or at a late cycle; defects discovered late are most likely be due to unclear specifications and are time consuming to resolve.	Medium	High	Defect management plan is in place to ensure prompt communication and fixing of issues.
<b>SCOPE</b> Scope completely defined	Medium	Medium	Scope is well defined but the changes are in the functionality are not yet finalized or keep on changing.
Natural disasters	Low	Medium	Teams and responsibilities have been spread to two different geographic areas. In a catastrophic event in one of the areas, there will resources in the other areas needed to continue (although at a slower pace) the testing activities.
Non-availability of Independent Test environment and accessibility	Medium	High	Due to non availability of the environment, the schedule gets impacted and will lead to delayed start of Test execution.

Delayed Testing Due To new Issues	Medium	High	<p>During testing, there is a good chance that some “new” defects may be identified and may become an issue that will take time to resolve.</p> <p>There are defects that can be raised during testing because of unclear document specification. These defects can yield to an issue that will need time to be resolved.</p> <p>If these issues become showstoppers, it will greatly impact on the overall project schedule.</p> <p>If new defects are discovered, the defect management and issue management procedures are in place to immediately provide a resolution.</p>
-----------------------------------	--------	------	---

## 8. GITHUB AUTOMATION SOURCE CODE REPOSITORY

<https://github.com/abhimenon77/CustomerOnboardingApplication>