

**Prepared by:**  
**BANDARI MAHESH**  
**TEST ENGINEER**

**Name Of the Document: TEST PLAN**  
**Application: Orange HRM (Open Source)**

## **TABLE OF CONTENTS**

### **1.0 INTRODUCTION**

- 1.1 Purpose
- 1.2 Project Overview

### **2.0 TEST STRATEGY**

- 2.1 Test Objectives
- 2.2 Test Assumptions
- 2.3 Test Principles
- 2.4 Data Approach
- 2.5 Scope and Levels of Testing
  - 2.5.1 Smoke Testing
  - 2.5.2 Sanity Testing
  - 2.5.3 Exploratory Testing
  - 2.5.4 Retesting
  - 2.5.5 Regression testing
  - 2.5.6 Functional Test
  - 2.5.7 User Acceptance Test (UAT)

### **3.0 Execution Strategy**

- 3.1 Entry and Exit Criteria
- 3.2 Test Cycles
- 3.3 Validation and Defect Management
- 3.4 Defect Tracking & Reporting
- 3.5 Test Management Process
- 3.6 Test Execution Process
- 3.7 Test Risks and Mitigation Factors

### **4.0 Testing Strategy (Subjected to Change)**

- 4.1 Features to be Tested
- 4.2 Features not to be tested
- 4.3 Major Deliverables
- 4.4 Tools

### **5.0 Approvals**

## **1.0 INTRODUCTION**

### **1.1 Purpose**

This test plan describes the testing approach and overall framework that will drive the testing of the Orange HRM-PIM module. This document introduces:

- Test Strategy: Rules the test will be based on, including the given project (objectives, assumptions) description of the process to setup a valid test (e.g.: entry/exit criteria, creation of test cases, specific task to perform, scheduling, data strategy).
- Execution Strategy: Describes how the test will be performed and process to identify and report defects, and to fix and implement fixes.

### **1.2 Project Overview**

- Detailed User Manual available in the repository which will contains all the do's that can be done with PIM module.

## **2.0 TEST STRATEGY**

### **2.1 Test Objectives**

The objective of the test is to verify that the functionality of Orange HRM -PIM module works according to specification.

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

### **2.2 Test Assumptions**

#### **General**

- Exploratory testing will be carried out once the build is ready for testing.
- All the defects would come along with a snapshot in either JPEG or JPG format (Video footage based on necessity).
- The test team assumes all the inputs wherever required will be given by Business Analyst/Developers.
- Test Case design activity will be performed by QA Team.
- Test Environment and preparation activity will be owned by Developers/Business Analyst.
- Dev Team will provide defect fixing plans based on the defect meeting during each cycle to plan.
- Business Analyst/Test Team lead/Project Manager will review and sign-off all the test cases prepared by Test Team prior to start of the test execution.
- Project Manager/Business Analyst will review and sign-off all test deliverables.

- Defects will be tracked through the GitHub only.
- Test Team will manage the testing effort with close coordination with Project Manager/Business Analyst/Developers.
- 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.
- Re & Regression testing will be carried out upon defects fixing confirmation received from developers.

### **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 performed Functional testing only on PIM Module.

### **UAT:**

UAT test execution will be performed by end users and QA Group will provide their support on creating UAT script.

## **2.3 Test Principles**

- Testing will be focused on meeting the business objectives, cost efficiency and quality.
- Testing process will be well defined, yet flexible, with ability to change as needed.
- Testing environment and data will emulate a production environment as much as possible.
- Testing will be repeatable, quantifiable, and measurable activity.
- There will be entrance and exit criteria.

## **2.4 Data Approach**

In functional testing, PIM Module will contain pre-loaded test data which is used for testing activities.

## **2.5 Scope and Levels of Testing**

Following list of testing comes under System/Functional testing which are under scope (While designing this test plan).

### **2.5.1 Smoke Testing**

**Purpose:** Smoke testing is a confirmation for QA team to proceed with further software testing.

**Testers:** Myself

**Method:** The test will be performed according to functional scripts, which are stored in the testing repository.

### **2.5.2 Sanity Testing**

**Purpose:** Sanity testing is performed to ensure that the code changes that are made are working properly.

**Testers:** Myself

**Method:** Sanity testing is unscripted form of testing. The testing team conducts basic tests which are focused on the new functionality or change and its impacts.

### **2.5.3 Exploratory Testing**

**Purpose:** The purpose of this test is to make sure critical defects are removed before the next level of testing can start.

**Testers:** Myself

**Method:** This exploratory testing is carried out in the application without any test scripts and documentation.

### **2.5.4 Retesting:**

**Purpose:** The purpose of this test is to make sure to check if the identified defects are fixed or not

**Testers:** Myself

**Method:** Retesting is done by replicating the same scenario with same data in new build

### **2.5.5 Regression testing**

**Purpose:** The purpose of this test is to make sure to find out whether the updates or changes had caused new defects in existing functions.

**Testers:** Myself

**Method:** Regression testing is performed by preparing the Regression suite.

### **2.5.6 Functional Test**

**Purpose:** Functional testing will be performed to check the functions of application. The functional testing is carried out by feeding the input and validates the output from the application.

**Testers:** Myself

**Method:** The test will be performed according to functional scripts, which are stored in the testing repository.

### **2.5.7 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 before deployment.

**Testers:** The UAT is performed by end users.

**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,L3) and Business analyst

### 3.0 Execution Strategy

#### 3.1 Entry and Exit Criteria

- The entry criteria refer to the desirable conditions in order to start test execution.
- The exit criteria are the desirable conditions that need to be met in order proceed with the exit from testing phase.
- Entry and exit criteria are flexible benchmarks. If they are not meet, 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.

Sl. No	Exit Criteria	Test Team (Myself)	Notes
01	100% Test scripts executed		
02	95% Pass rate of Test Scripts		
03	No Open Critical and Severity bugs		
04	All remaining defects are either cancelled or documented as Change Requests for a future release		
05	95% of medium severity defects have been closed		
06	All Expected and actual results are captured and documented with test script		
07	All defects logged in GitHub		
08	Test Closure Memo completed and signed off		

#### 3.2 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 and blocking, critical defects, and most of the high defects. It is expected to use some work around in order to get all the scripts.
- The Objective of the second cycle is to identify remaining high and medium defects, remove the work around the first cycle, correct gaps in the scripts and obtain performance results.
- UAT test will consist of one cycle.

### 3.3 Validation and Defect Management

- It is expected that 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.
- The defects will be tracked through GitHub only.
- 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 Test Lead/Project 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 progress through the cycle; it is the responsibility of the technical team to review and ask for details if necessary , fix the defect, communicate to the Test Lead fix is done.

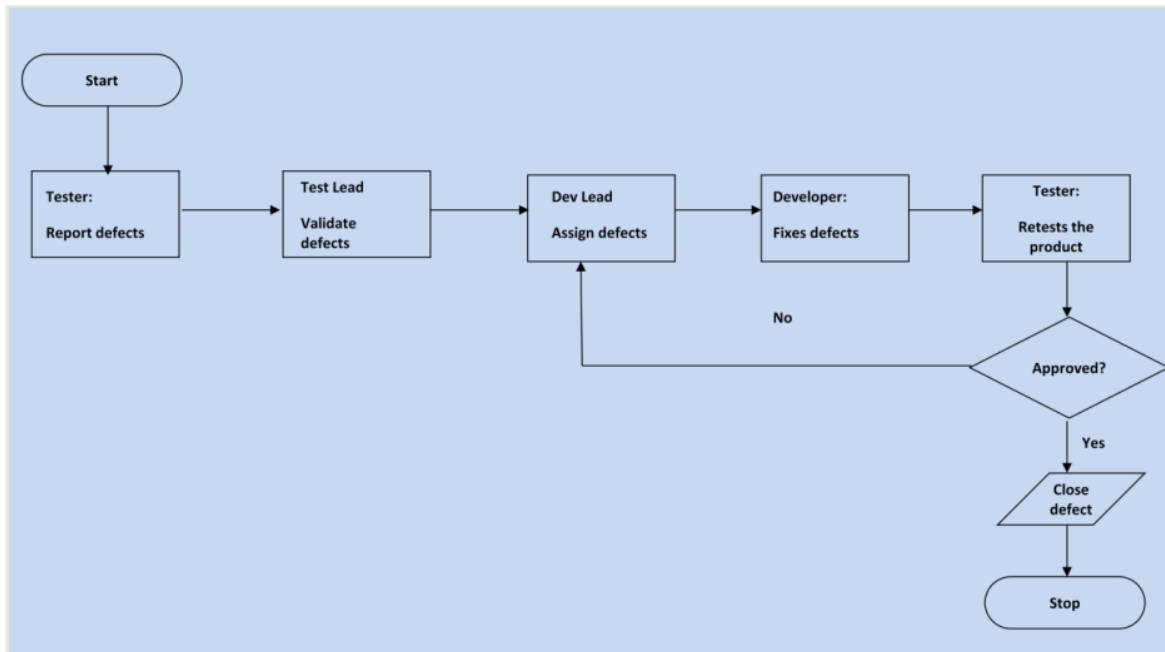
Defects found during the testing will be categorized to the bug-reporting tool “GitHub” and categories are:

Severity	Impact
1.Critical	The bug is critical enough to crash the system, cause file corruption or cause potential data loss. It causes an abnormal return to the operating system.
2.High	It causes a lack of vital program functionality with workaround.
3. Medium	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. This bug prevents other areas of the product from being tested. However, other areas can be independently tested.
4. Low	There is insufficient or unclear error message, which has minimum impact on product use.
5. Cosmetic	There is an insufficient or unclear error message that has no impact on product use.

### 3.4 Defect Tracking & Reporting

Following flow chart depicts Defect Tracking process:





### 3.5 Test Management 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 Requirements as part of Traceability matrix.
- Each of the test cases will undergo review by Test Lead 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.

### 3.6 Test Execution Process

- Once all test cases are approved and test environment is ready for testing, tester will start a smoke test of the application to ensure application is stable for testing.
- Each tester is assigned test cases directly.
- Testers to ensure necessary access to the testing environment, GitHub for raising defects. If any issues will be escalated to the Test Lead and in turn to the Project Manager as escalation.
- Each tester performs step by step execution and update execution status. The tester enters Pass or Fail Status.
- If any failures, defects will be raised as per severity guidelines detailing steps to simulate along with screenshots if appropriate.
- This process is repeated until all test cases are executed full with Pass/Fail status.

### 3.7 Test Risks and Mitigation Factors

<b>Risk</b>	<b>Probability</b>	<b>Impact</b>	<b>Mitigation Plan</b>
1. 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	The testing team can control the preparation tasks (in advance) and early communication with involved parties. Some buffer has been added to the schedule for contingencies although not as much as best practices advice.
2. Not enough time, project/application development is too late	Medium	High	Extra working hours have been estimated and built into the schedule.
3. 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 issues.

### 4.0 TEST ENVIRONMENT

A windows OS with Mozilla Firefox 104.0 onwards as well as Chrome 99.0 and later should be available to each tester.

#### 4.1 FEATURES TO BE TESTED

- OrangeHRM-PIM Module

#### 4.2 FEATURES NOT TO BE TESTED

- Admin Module,
- Time Module
- Recruitment Module
- Performance Module

#### 4.3 Major Deliverables

- Test Plan
- Test Cases and Test Scenario documents.
- Test Summary Reports.
- Bug Report

**Prepared by:**  
**BANDARI MAHESH**  
**TEST ENGINEER**

#### **4.4 Tools**

Automation Tool: Selenium 4.0 with Java.

Bug Tracking Tool: Github

#### **5.0 APPROVALS**

<b>Name (In Capital Letters)</b>	<b>Signature</b>	<b>Date</b>
1.		
2.		
3.		
4.		