Test Plan

03/09/2023

**Name of the tester:**

**Version:**1.0

**Created:** 03/09/2023

**Last Updated:**

**Status:** DRAFT

# Revision and Signoff Sheet

**Document History**-

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description of Change** |
| 1 | 03/09/2023 | Vinutha | Draft |
|  |  |  |  |

**Approvers List**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Role | Approver /  Reviewer | Approval /  Review Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Table of Contents

Table of Contents

[Revision and Signoff Sheet 2](#_Toc144734881)

[Table of Contents 3](#_Toc144734882)

[1. INTRODUCTION 5](#_Toc144734883)

[1.1. Purpose 5](#_Toc144734884)

[1.2. Project Overview 5](#_Toc144734885)

[1.3. Audience 5](#_Toc144734886)

[2. TEST STRATEGY 6](#_Toc144734887)

[2.1. Test Objectives 6](#_Toc144734888)

[2.2. Test Assumptions 6](#_Toc144734889)

[Key Assumptions 6](#_Toc144734890)

[General 6](#_Toc144734891)

[Functional Testing 7](#_Toc144734892)

[2.3. Test Principles 7](#_Toc144734893)

[2.4. Data Approach 7](#_Toc144734894)

[2.5. Scope and Levels of Testing 7](#_Toc144734895)

[2.5.1. Functional Test 7](#_Toc144734896)

[TEST ACCEPTANCE CRITERIA 8](#_Toc144734897)

[TEST DELIVERABLES 8](#_Toc144734898)

[MILESTONE LIST 9](#_Toc144734899)

[2.5.2. User Acceptance Test (UAT) 9](#_Toc144734900)

[2.5.3. Automation 9](#_Toc144734901)

[TEST ACCEPTANCE CRITERIA 10](#_Toc144734902)

[2.6. Test Effort Estimate 11](#_Toc144734903)

[3. EXECUTION STRATEGY 11](#_Toc144734904)

[3.1. Entry and Exit Criteria 11](#_Toc144734905)

[3.2. Test Cycles 12](#_Toc144734906)

[3.3. Validation and Defect Management 12](#_Toc144734907)

[3.4. Test Metrics 13](#_Toc144734908)

[3.5. Defect tracking & Reporting 14](#_Toc144734909)

[4. TEST MANAGEMENT PROCESS 14](#_Toc144734910)

[4.1. Test Management Tool 14](#_Toc144734911)

[4.2. Test Design Process 16](#_Toc144734912)

[4.3. Test Execution Process 17](#_Toc144734913)

[4.4. Test Risks and Mitigation Factors 17](#_Toc144734914)

[5.2. Role Expectations 19](#_Toc144734915)

[5.2.1. Project Management 19](#_Toc144734916)

[5.2.2. Test Planning (Test Lead) 19](#_Toc144734917)

[5.2.3. Test Team 21](#_Toc144734918)

[5.2.4. Test Lead 21](#_Toc144734919)

[5.2.5. Development Team 21](#_Toc144734920)

[5. TEST ENVIRONMENT 21](#_Toc144734921)

[6. Non functional Testing 21](#_Toc144734922)

[6.1. Test Objectives: 21](#_Toc144734923)

[6.2. Scope: 22](#_Toc144734924)

[6.3. Out of Scope: 22](#_Toc144734925)

[6.4. Approach: 22](#_Toc144734926)

[6.4.1Performance Metrics 22](#_Toc144734927)

[6.5 Entry and Exit Criteria: 23](#_Toc144734928)

[6.5.1 Entry Criteria: 23](#_Toc144734929)

[6.5.2 Exit Criteria: 23](#_Toc144734930)

[6.6 Suspension and Resumption Criteria: 23](#_Toc144734931)

[6.7 Test Deliverables: 23](#_Toc144734932)

[6.8 Risk and Mitigation plan: 23](#_Toc144734933)

[6.9 Assumptions: 24](#_Toc144734934)

[6.10 Dependencies: 24](#_Toc144734935)

[7. APPROVALS 24](#_Toc144734936)

# INTRODUCTION

## Purpose

This test plan describes the testing approach and overall framework that will drive the testing of the Unicourt Sales Portal version 1.0– unicourt.com site. The document introduces:

* + - Test Strategy: rules the test will be based on, including the givens of the project (e.g.: start / end dates, objectives, assumptions); description of the process to set up a valid test (e.g.: entry / exit criteria, creation of test cases, specific tasks 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.
    - 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)

## Project Overview

Unicourt Support portal is a powerful tool providing employees of Unicourt with the ability to view relevant modules such as Dashboard, Accounts, Contracts, Reports and Admin based on the privilege set to the authenticated user.

The Sales and Support team members work from different locations, the support portal shall be accessible from a any location through web browser.

Based on the privilege the authenticated user should be able to view/edit the modules.

## 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.
    - 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.

# TEST STRATEGY

## Test Objectives

The objective of the test is to verify that the functionality of Unicourt Support Portal 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.

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.

## Test Assumptions

### Key Assumptions

* + - Users with username containing unicourt.com and having different privileges being created.
    - 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

* + - Smoke and Exploratory Testing would be carried out once the build is ready for testing
    - Performance testing will be performed on the stable product.
    - 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 test management tool(Name of the tool used in Unicourt). 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.

### Functional Testing

* + - During Functional testing, testing team will use preloaded data which is available on the system at the time of execution

**Automation Testing**

* + - Separate automation environment should be available for the automation scripts to execute. It should have daily latest build. The build should be stable and necessary test data should be created before script execution.

## 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.
    - There will be entrance and exit criteria.

## Data Approach

* + - In functional testing, Unicourt Support Portal will contain users created with @unicourt domain and assigned with different roles(Viewer, Sales, Support, Admin)

## Scope and Levels of Testing

### 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.

**Scope:** Test cases reviewed and approved by the PM will be executed through the test management tool and the status will be updated accordingly. In case of failures, the failed test cases will be re-executed after the fix.

**TESTERS**: Testing Team.

**TIMING**: after Smoke test is completed.

### TEST ACCEPTANCE CRITERIA

* + - 1. Approved Functional Specification document, Use case documents must be available prior to start of Test design phase.
      2. Test cases approved and signed-off prior to start of Test execution
      3. Development completed, unit tested with pass status and results shared to Testing team to avoid duplicate defects
      4. Test environment should be in a ready to use state.
* Approved Functional Specification Document
* Approved Use cases
* Approved Test cases

Sign-off

* Development completed & unit tested
* Application deployed and system ready for testing on Test environment
* Production like data is available to test all functionalities.
* Defect fixes planned based on Defect triage (Unit Testing) and evaluation criteria

Readiness

### TEST DELIVERABLES

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | Test Team | Test Lead/  Dev  Lead |
| 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

1. Any issues in the System environment readiness
2. Any change in scope/addition in scope
3. Any other dependency that impacts efforts and timelines

### User Acceptance Test (UAT)

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

**TESTERS**: the UAT is performed by the Sales/support team

**METHOD**: Since the Sales/Support users are the going to use this application, they test the application, verify and validate that the application is working as per the business needs.

**TIMING**: After QA provides Sign off in Test environment and before deploying the code to Production.

### Automation

**PURPOSE**: this test focuses on automating the user stories and maintaining the existing automation scripts.

**TESTERS**: Assigned testers automate the user stories which they verify manually.

**METHOD**: Each QA will automate the user stories which they are verifying manually after analyzing the feasibility. Playwright with JavaScript will be used to automate..

**TIMING**: After QA completes user story veification and once the functionality is stable, automation script will be developed.

### TEST ACCEPTANCE CRITERIA

* + - 1. Testing team will identify the test cases needs to be automated.
      2. Sanity and regression suite both will be run and should be passed with minimum 90% pass rate.
      3. Any defects identified through automation must be fixed within the same release.
      4. Separate Test environment should be available for the automation and it should have daily latest build same as of the QA/Manual test environment.

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Deliverable Name** | **Author** | **Reviewer** |
| 1. | Test Plan | Test Lead | Project Manager/  Business Analyst’s |
| 2. | Automation Test Cases | Test Team | Project Manager/  Business Analyst’s |
| 3. | Execution Report | Test Team | Test Lead/ Project Manager |
| 4. | Daily/weekly status report | Test Team/ Test Lead | Test Lead/ Project  Manager |

# EXECUTION STRATEGY

## 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** | **Technical**  **Team** | **Notes** |
| 100% Test cases executed |  |  |  |
| 95% pass rate of Test cases |  |  |  |
| No open Critical and High severity defects |  |  |  |
| 95% of Medium severity defects have been closed |  |  |  |
| All remaining defects are either cancelled or  Moved to the future release with proper reason and as a known defect |  |  |  |
| All the stories have been automated |  |  |  |
| 100% automation test scripts executed |  |  |  |
| 90% pass rate of automation test scripts |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| All expected and actual results are captured and  documented with the test script |  |  |  |
| All test metrics collected based on reports from test management tool |  |  |  |
| All defects logged in JIRA |  |  |  |
| Test Closure Memo completed and signed off |  |  |  |
| Test environment cleanup completed and a new  back up of the environment |  |  |  |

## Test Cycles

* There will be two cycles for functional testing. Each cycle will execute sanity and regression suite.
* Cases which are not automated or not feasible to automate will be tested manually.
* 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.
  + - UAT test will consist of one cycle.

## Validation and Defect Management

* + - It is expected that the testers execute all the testcases 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 testcases. This is especially relevant in the second cycle, when the Business analyst’s/PM/Support team join in the execution of the test, since the BUSINESS ANALYSTs/PM/Support team have a deeper knowledge of the business processes. If a gap is identified, the testcases and traceability matrix will be updated and then a defect logged against the testcases.
    - The defects will be tracked through JIRA only. The technical team will gather information on a daily basis from JIRA, and request additional details from the tester. The technical team will work on fixes.
    - It is the responsibility of the tester to open the defects, link them to the corresponding testcases, assign an initial severity and status, retest and close the defect; it is the responsibility of the Manager to review the severity of the defects and facilitate with the technical team to 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 JIRA on a daily basis, ask for details if necessary, fix the defect, communicate to the Manager the fix is done, implement the solution per the Manager request.
    - Customer reported defects will be analyzed and based on the review the scenario will be added to the automation suite.

Defects found during the Testing will be categorized according to the bug-reporting tool “JIRA” and the categories are:

|  |  |
| --- | --- |
| **Severity** | **Impact** |
| 1 (Critical) | * This 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 (crash or a system failure message appears). * It causes the application to hang and requires re-booting the 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 an 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. |

## Test Metrics

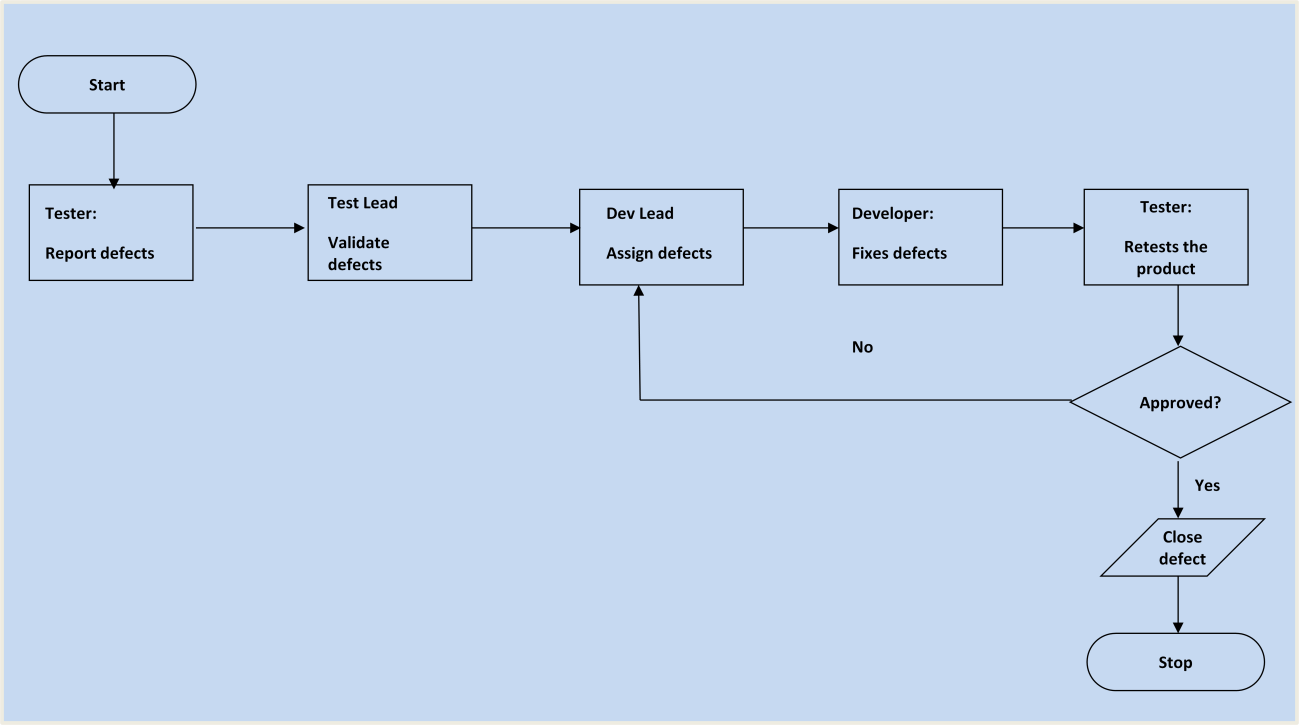
Test metrics to measure the progress and level of success of the test will be developed and shared with the project manager for approval. The below are some of the metrics

|  |  |  |
| --- | --- | --- |
| **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) |
| Sprint wise status report | Status report on manual test case execution, defects identified, reopened, closed, open, inprogress.open, inprogress.  Automation execution report |  |
| Project | Project driven reporting (As requested by PM) | Weekly – If project |
| Weekly |  | team needs weekly |
| Status |  | update apart from |
| report |  | daily and there is |
|  |  | template available |

|  |  |  |
| --- | --- | --- |
|  |  | with project team to use. |

## Defect tracking & Reporting

Following flowchart depicts Defect Tracking Process:



# TEST MANAGEMENT PROCESS

## Test Management Tool

TestRail is the tool used for Test Management. All testing artifacts such as Test cases, test results are updated in the Testrail.

* + - Project specific folder structure will be created in Testrail to manage the status of this DFRT project.
    - Sprint milestones will be created and the test cases specific to each sprint will be added and executed in the sprint
    - Each resource in the Testing team will be provided with Read/Write access to add/modify Test cases in Testrail.
    - During the Test Design phase, all test cases are written directly into Testrail. Any change to the test case will be directly updated in the Testrail.
    - Each Tester will directly access their respective assigned test cases and update the status of each executed step in Testrail directly.
    - Any defect encountered will be raised in JIRA 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 Testrail.
    - Various reports can be generated from Testrail to provide status of Test execution. For example, Status report of Test cases executed, Passed, Failed, No. of open defects, Severity wise defects etc.
    - Once the test cases assigned to sprint are executed and sprint is closed, the specific sprint milestone will be closed.

## Test Design Process

Understanding

Requirements

Establishing

Traceability Matrix in Testrail

Incorporating

Preparation of SME /Peer Review st cases Review of Test comments in

cases test cases

* + - 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 Testrail.

## Test Execution Process

Execute each of

the test step in test case

Mark Status as Raise defects for Send the daily

Pass/Fail in

Testrail the failed test status report cases in HP ALM Test Lead

Participate in

Defect Triage cycle and explain the defects

Complete the

test execution of all the test cases

* + - Once all Test cases are approved and the test environment is ready for testing, tester will start a smoke test of the application to ensure the application is stable for testing.
    - Each Tester is assigned Test cases directly in Testrail.
    - Testers to ensure necessary access to the testing environment, Testrail 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 smoke testing will be escalated to the respective development SPOCs 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 Testrail.
    - If any failures, defect will be raised as per severity guidelines in JIRA tool detailing steps to simulate along with screenshots if appropriate.
    - Daily Test execution status as well as Defect status will be communicated in the scrum with

Stake holders.

* + - 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 Testrail 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 Testrail during the cycle.

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

## Test Risks and Mitigation Factors

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Prob.** | **Impact** | **Mitigation Plan** |
| **SCHEDULE**  Testing schedule is tight. If the start | High | High | * The testing team can control the preparation tasks (in advance) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Prob.** | **Impact** | **Mitigation Plan** |
| of the testing is delayed due to design tasks, the test cannot be extended beyond the UAT scheduled start date. |  |  | and the early communication with involved parties.   * Some buffer has been added to the schedule for contingencies, although not as much as best practices advise. |
| **RESOURCES**  Not enough resources, resources on boarding too late | Medium | High | Holidays and vacation have been estimated and built into the schedule; deviations from the estimation could derive in delays in  the testing. |
| **DEFECTS**  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. |
| **SCOPE**  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 | 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.  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.  If these issues become  showstoppers, it will greatly impact |

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Prob.** | **Impact** | **Mitigation Plan** |
|  |  |  | on the overall project schedule.  If new defects are discovered, the defect management and issue management procedures are in place to immediately provide a resolution. |
| Automation environment issue | High | High | If the automation environment is not available due to build issue or if the latest build is not available in the automation environment, this will impact the regression cycle and can impact the schedule. |

* 1. **Communications Plan and Team Roster**

## Role Expectations

The following list defines in general terms the expectations related to the roles directly involved in the management, planning or execution of the test for the project.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Roles** | **Name** | **Contact Info** |
| 1. | Project Manager |  |  |
| 2. | Test Lead |  |  |
| 3. | Business Analyst |  |  |
| 4. | Development Lead |  |  |
| 5. | Testing Team |  |  |
| 6. | Development Team |  |  |

### Project Management

* + - * Project Manager: reviews the content of the Test Plan, Test Strategy and Test Estimates signs off on it.

### Test Planning (Test Lead)

* + - * Ensure entrance criteria are used as input before start the execution.
      * Develop test plan and the guidelines to create test conditions, test cases, expected results and execution scripts.
      * Provide guidelines on how to manage defects.
      * Attend status meetings in person or via the conference call line.
      * Communicate to the test team any changes that need to be made to the test deliverables or application and when they will be completed.
      * Provide on premise or telecommute support.

### Test Team

* + - * Develop test conditions, test cases, expected results, and execution scripts.
      * Perform execution and validation.
      * Identify, document and prioritize defects according to the guidance provided by the Test lead.
      * Re-test after software modifications have been made according to the schedule.
      * Prepare testing metrics and provide regular status.

### Test Lead

* + - * Acknowledge the completion of a section within a cycle.
      * Give the OK to start next level of testing.
      * Facilitate defect communications between testing team and technical / development team.

### Development Team

* + - * Review testing deliverables (test plan, cases, scripts, expected results, etc.) and provide timely feedback.
      * Assist in the validation of results (if requested).
      * Support the development and testing processes being used to support the project.
      * Certify correct components have been delivered to the test environment at the points specified in the testing schedule.
      * Keep project team and leadership informed of potential software delivery date slips based on the current schedule.
      * Conduct first line investigation into execution discrepancies and assist test executors in creation of accurate defects.
      * Implement fixes to defects according to schedule.

# TEST ENVIRONMENT

* + - * Environment with Firefox and Chrome browser should be available to each tester.
      * Automation environment should have the daily latest build.

# Non functional Testing

## Test Objectives:

* + - * To ensure that the application performance, service availability, and the stability of the application are not impacted as a result of the enhancements.
      * Response Times remain within the acceptable tolerance over the increasing Load.
      * Verify stable memory usage over the increasing load.

## Scope:

* + - * Identify the critical scenarios for performance testing with assistance from different project tracks.
      * Preparation of performance test scripts using JMeter for the identified scenarios that mimic the identified peak load.
      * Publish Performance test results.
      * Coordinate with various stakeholders to resolve the identified performance issues.
      * Baseline the performance level for future releases.

## Out of Scope:

* [Functional Testing](https://www.softwaretestinghelp.com/guide-to-functional-testing/), UAT, System Testing & Security Testing.
* Performance testing/monitoring of any third-party interfaces.

## Approach:

* The test scenario will be scripted using the JMeter tool. The scripts would be customized as required. The schedule will be created with the required ramp-up to simulate the real-world scenarios.
* The Test Scenario would be broken up and measured in the below aspects:
* **Baseline Test:** To run each scenario with 1 Vuser and multiple iterations in order to identify whether the application performance meets the business Service Level Agreement or not.
* **Base Load Test:** To meet the Business Benchmark under load test, the Testing team will perform a baseload test which will help to identify any system performance issues with increasing load and creates the baseline for the next level of performance testing.
* **Peak Load / Scalability Test:**  Testing team will perform multiple tests with increasing Vusers to meet the expected load and also to measure the application performance to establish the performance curve and identify whether the deployment can support the service level agreements under the peak user load(increasing the no of Vusers to 50%, 75%, 100% and 125% of peak capacity).
* **Endurance Test:** Testing team will run this test for a period of 8 Hours / 16 Hours /24 hours to identify memory leaks, performance issues over time, and overall system stability. During endurance tests, the Testing team monitors the key performance indicators, such as transaction response times and the stability of memory usage.
* System resources like CPU, Memory, and IO need to be monitored with the help of the project team.

## 6.4.1Performance Metrics

**Client-side Metrics**

| **S.No** | **Metric** | **Description** |  |
| --- | --- | --- | --- |
| 1 | Transaction Response Time | Response time of pages during the steady state of the performance test |  |
| 2 | Throughput | The amount of data that the VUsers received from the server over time |  |
| 3 | Hits/second | The number of HTTP requests made by VUsers to the Web server during the scenario run |  |
| 4 | Number of Passed/Failed Transactions | Total number of transactions that Passed and Failed during the test execution |  |
| 5 | Transaction Error Rate | The Percentage of transactions that failed during the test execution |  |

## Entry and Exit Criteria:

## Entry Criteria:

* Access to all the applications in the environment.
* Environment readiness complete.
* Performance Test Data readiness.

## Exit Criteria:

* 100% in scope test cases are executed and results are documented.
* No Critical defects.
* Any defects deferred have approval and documented work around.
* The Support portal shall be available with uptime of 99.999%
* The portal should serve 100 users connected at a time
* The portal shall serve 50 page requests per second

## Suspension and Resumption Criteria:

| **Suspension** | **Impact** | **Resumption** |
| --- | --- | --- |
| Environment not set up | Testing cannot proceed | Environment readiness. |
| Application found to be unstable | Testing cannot proceed. | Issue resolved |
| Test Data not available | Testing cannot proceed. | Test Data ready |

## Test Deliverables:

* Performance Testing Strategy
* Performance Test Scenario Document
* Performance Test Scripts
* Performance Test Results

## Risk and Mitigation plan:

|  |
| --- |
|  |
| **S.No** | **Risk** | **Probability** | **Impact** | **Mitigation Plan** | **Owner** |
| 1 | Test Data unavailability for performance load test executions | H | H | Estimated dates for the performance test executions should be reviewed and updated. Functional/Dev team support required for data gathering. | -- |
| 2 | Environmental Issues | L | M | Re prioritize Deliverables | -- |
| 3 | Change in Functionality/design during performance test execution | M | H | This requires rework on the performance test scenarios | -- |
| 4 | Extra performance runs to troubleshoot performance issues | M | H | Performance testing schedules would be modified and updated to the product team. | -- |
| 5 | If the defects gets reopened multiple times or fixing the defect causes performance dip | H | H | Re prioritize the performance test execution cycles. | -- |
| 6 | Hardware Availability | M | H | Schedule start date would be moved accordingly. |  |

## Assumptions:

* Performance Test Environment will be a replica of the product architecture landscape. (i.e. correct Hardware, Software, Interfaces, Integration Layers, etc).
* Performance scripts will be designed based on the critical flows for which the usage is high.
* All Infrastructure Issues should be resolved before the beginning of Performance testing. Any system configuration changes made later will invalidate the test results.
* An application is stable and ready to use in the Performance test environment.
* Necessary hardware and software resources (like load generator machines/software, controller/agent machines) are made available.
* Any changes to the scope will go through a change control process and the testing team will assess the impact of timelines and resources.
* Respective Servers are expected to handle the load.
* Application trace logs have to be enabled for the supporting systems for monitoring purposes.

## Dependencies:

* Availability of the Performance test environment which is a replica of the product architecture landscape.
* Support required from various Functional, Development, Database and Infrastructure teams during the test preparation and execution stages.
* No code changes are implemented during the entire Performance testing phase as time is very limited.
* In the event of unforeseen issues that lead to restrictions within the timelines, if timelines do not allow for all the test scopes to be met within the original milestone dates support is available from the Release Managers, to provide a scoping and prioritization decision.
* Application Business Users / Subject Matter Experts will be made available for functional clarifications, and business transactions sign-off.
* Manager will review and sign-off.

# APPROVALS

The Names and Titles of all persons who must approve this plan.

|  |  |
| --- | --- |
| **Signature:** |  |
| **Name:** |  |
| **Role:** |  |
| **Date:** |  |

|  |  |
| --- | --- |
| **Signature:** |  |
| **Name:** |  |
| **Role:** |  |
| **Date:** |  |