

# TEST PLAN

## 1. TEST INTRODUCTION

### 1.1. Purpose

This test plan describes the testing approach and overall framework that will drive the testing of the Hive user onboarding for brand new users. The document introduces:

- Test Strategy: rules the test will be based on, description of the process to set up a valid test.
- 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

Hive is the insanely productive new tool for teams. It puts all actions, messages and files on one simple page. The flow of new user signup is the most important one, it's a business critical flow and failure here kills confidence in the product for users.

## 2. TEST STRATEGY

### 2.1. Test Objectives

The objective of the test is to verify that the new users can successfully join Hive.

### 2.2. Test Assumptions

- User information required will be available in the system prior to start of testing.
- Testing will be focused on new user signup flow.

### 2.3. Test Principles

- Testing will be a repeatable, quantifiable, and measurable activity.
- Testing will be divided into distinct phases, each with clearly defined objectives and goals.

### 2.4. Scope and Levels of Testing

#### 2.4.1. Exploratory

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

**TESTERS:** Testing team.

**METHOD:** this exploratory testing is carried out on <https://staging.hive.com> without any test scripts

**TIMING:** at the beginning of each cycle.

#### 2.4.2. Functional Test

**PURPOSE:** Functional testing will be performed to test user onboarding flow. The functional testing is carried out by feeding user information and clicking buttons and validates the output webpage from the signup page.

**TESTERS:** Testing Team.

**METHOD:** The test will be performed according to test scripts.

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

#### **TEST ACCEPTANCE CRITERIA**

1. Test cases approved prior to start of test execution
2. Development completed, unit tested with pass status and results shared to testing team to avoid duplicate defects
3. Test environment with application installed, configured and ready to use state

#### **TEST DELIVERABLES**

| S.No. | Deliverable Name           | Author               | Reviewer                               |
|-------|----------------------------|----------------------|--|
| 1.    | Test Plan                  | Test Lead            | Project Manager/<br>Business Analyst's |
| 2.    | Functional Test Cases      | Test Team            | Business Analyst                       |
| 3.    | Logging Defects            | Test Team            | Test Lead/<br>Programming Lead         |
| 4.    | Daily/weekly status report | Test Team/ Test Lead | Test Lead/ Project<br>Manager          |
| 5.    | Test Closure report        | Test Lead            | Project Manager                        |

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

**PURPOSE:** this test focuses on validating the business logic and webpage function. 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.

**METHOD:** Since Hive users are the most indicated to provide input around teamwork needs and how the application 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.

**TIMING:** After exploratory and functional testing 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 |

### **3. 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 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.

| Exit Criteria  | Test Team | 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 |           |       |

#### **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 any critical defects, and most of the high defects. The objective of the second cycle is to identify remaining high and medium defects, remove the them and obtain performance results. And the UAT test will consist of one cycle.

#### **3.3. 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 joins in the execution of test, since the business analyst have a deeper knowledge of the

business processes. If a gap is identified, the scripts will be updated and then a defect logged against the scripts.

- The defects will be tracked through testing. The technical team will gather defects and work on fixes.

Defects found during the Testing will be categorized as follow:

| Severity     | Impact   |
|--------------|--|
| 1 (Critical) | <ul style="list-style-type: none"><li>• This bug is critical enough to crash the system, cause file upload corruption, or cause potential data loss</li><li>• It causes the application to hang and requires re-booting.</li></ul>                   |
| 2 (High)     | <ul style="list-style-type: none"><li>• It causes a lack of vital program functionality.</li></ul>   |
| 3 (Medium)   | <ul style="list-style-type: none"><li>• This bug will degrade the quality of the System. However, there is a workaround for achieving the desired functionality.</li><li>• This bug prevents other areas of the product from being 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>  |

### 3.4. Defect tracking & Reporting

