|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Author** | Reviewed/Approved by | Comments | Approval Date |
|  |  |  |  |  |

1 objective 2

2 scope 3

3 Test methodology 2

4 test approach 4

5 assumptions 5

6 risk 6

7 backup plan/mitigation plan/contingency plan 3

8 roles and responsibilities 4

9 scheduling 4

10 defect tracking 5

11 test environment/test bed 4

10 defect tracking 4

12 entry and exit criteria 4

14 deliverables 5

13 test automation 4

14 deliverables 4

15 Release notes 6

# Objective

Test planing at each phase of STLC includes test strategies, features, Test methologies, Test approaches, risk, roles and responsibilities, and defect tracking process used for the product.

# Scope

### Here, we mention the features to be tested and features which not to be tested.

Modules and Features will be included in this product version are,

User Journey

Basic Search Home Page

Search Builder

Search History

Search Results Page

Author Finder

Personal Library

Journal Finder-Landing Page

Journal Finder-TOC Page

Footer

### Here, we mention what need not to be tested

Problems related to the contents of the “About”, “FAQ”, “Contact Us” Of the product Diskover

Problems related to the content of the journals

Need not to be tested as it has lowest risk and working well in the previous version of the product 2.0

# Test Methodology

## Overview

Based on the type of application and it’s functionalities, we decide what type of testing needs to be done.

* Web application
* Smoke testing
* Functional testing
* Integration testing
* System testing
* Compatibility testing
* Regression and Retesting
* UAT
* Mobile application
* Smoke testing
* Functional testing
* Integration testing
* System testing
* Compatibility testing
* Regression and Retesting
* UAT

# Test Approach

Here, we explain how we go about and test the application.

We have several approaches, such as

* Writing test scenarios
* Writing test cases
* Test case excecution using testlink
* Preparing test summary report

# Assumptions

Here, while writing the test plan, will assume a few things.

* Assumptions from a resource point of view
* Assumptions from the technology point of view
* Assumptions from supporting document point of view
* Assumptions from KT point of view

# Risks

When assumption fails then we will be in risk, so to overcome it we will have backup plans.

# Backup plan/Mitigation/Contigency plan

If assumptions fails, we will be in risk and it will lead to 100% risk to reduced to 20% we should have a backup plans.

# Roles and Responsibilities

* Test Manager
* He/She will review and approve the test plan.
* Interacting with development team and testing team and management team.
* Handling issue and escalations
* Involved in effort estimation and resourcing
* Approving the test case document
* Having knowledge on the domain and the product about business workflow, and guiding the team.
* Test engineer
* Understanding the product requirement
* Logging to the defect tracking tool(Assembla)
* Writing test scenarios and test cases using testlink
* Excecuting the test cases
* Performing functional and non functional testing
* Test summary report preperation

# Scheduling

In this section will mention start date and end date for each and every testing activities.

# Defect tracking

In order to report the bugs, assemble tool will be used

Procedure to track the defect

* Severity as follows below

1. Blocker
2. Critical
3. Major
4. Minor

* Priority as follows below
* Highest (1)
* High (2)
* Normal (3)
* Lowest (4)
* Low (5)

# Test Environment/Test bed

# Entry and Exit Criteria

# Test automation

All the modules and features needs to be tested manually

# Test Deliverables

These are the outputs or outcomes of testing team

* Test plan document
* Test scenarios and test case document
* Test case execution report/document
* Defect report document
* Release notes

# Release Notes

These are the set of document given to the customer along with the software/product which is signed off by the test manager.

* Version of software
* Procedure to use the product