**Buggy Software Web Application**

**Project:** Buggy Software

Test Strategy

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Description |
| 25-04-2023 | 0.0.1 | Ganesh S | Automation Framework first release |
|  |  |  |  |

Table of Contents

[1. Scope 3](#_Toc534636696)

[2. Test Approach 4](#_Toc534636697)

[3. Test Environment 5](#_Toc534636698)

[4. Testing Tools 6](#_Toc534636699)

[5. Release Control 7](#_Toc534636700)

[6. Risk Analysis 8](#_Toc534636701)

[7. Review and Approvals 9](#_Toc534636702)

# 

# Scope

* This application will be built in phases
* **Day 1:** Functional/smoke testing on the application
* **Day 2:** Find bugs at early stages and get them fixed
* **Document Approver:** James

# Test Approach

* Functional Testing will be carried out with basic health check up by smoke testing
* **Phase 1:** Functional/Smoke Testing
* **Phase 2:** Identify the automation scope
* **Phase 3:** Prepare unit tests and start the automation
* **Phase 4:** Execute the regression test using framework
* **Roles and Responsibilities:**
  + **Tester:** Ganesh S
  + **Types of Testing:** Functional and automation testing
  + **Roles & Responsibilities:** Perform Functional/Smoke Testing, Find as many bugs at early stages and get them fixed, build a framework and perform regression.
* **Automation Tools:** Selenium Webdriver, Java, BDD, cucumber, Maven, Junit
* Adding new defects, re-testing, Defect triage, Regression Testing and test sign off

# 

# Test Environment

* **Platform:** Windows, Java
* **Automation Tools & methodology:** Selenium Webdriver, BDD, cucumber, Maven, junit
* **Repository or version control:** GIT

# Testing Tools

* Automation and Test management tools needed for test execution

Selenium webdriver, Cucumber, Java, junit, Maven

* We will use most of the opensource software/applications. In case any licensed version required will be investigated and planned accordingly.

# 

# Release Control

* Maximum Test Coverage
* All the bugs are identified and get it fixed at early stages
* Maintain proper code using version control
* Make sure all the tests are executed and well tested
* Prepare release document along with Test Evidences.
* Identify the deferred bugs and make a note of it.

# Risk Analysis

* + Dependencies’ versions (In automation framework) are stable before release.
  + Timely communication with client for the status and negative events
  + Identify the critical bugs at early stages and if skipped then plan those accordingly to avoid the risk.
  + Maximum use of open source and try to complete the testing in time.
  + Backup plan in resource’ absence for the skilled resource.
  + Timely deliveries and client’s early feedbacks.

# 

# Review and Approvals

* All these activities are reviewed and sign off by the business team, project management, development team, etc.
* Test evidences are attached, release notes are updated, Latest code has been merged with proper approvals.