drata TEST PLAN

**Change history**

|  |  |  |  |
| --- | --- | --- | --- |
| Version number | Date | Contributor | Description |
| 1.0 | 23/08/2022 | Irving Calzada | Document creation |

Index

Content

[Introduction 3](#_Toc95120688)

[Approach 4](#_Toc95120689)

[Scope 4](#_Toc95120690)

[Quality Objectives 5](#_Toc95120691)

[Roles and responsibilities 5](#_Toc95120692)

[Assumption for running tests 6](#_Toc95120693)

[Testing Methodology 6](#_Toc95120694)

[Suspension and resumption criteria 8](#_Toc95120695)

[Deliverables of tests 8](#_Toc95120696)

[Metrics 9](#_Toc95120697)

[Defect report 9](#_Toc95120698)

# Introduction

This document describes the test strategies defined for the DRATA project. Processes and plans for problem resolution or contingences will also be covered if necessary.

# Approach

DRATA is an extensive project that will cover certain needs that the client requires in the project process:

* Regression tests.
* Smoke Tests.
* Acceptance Tests.
* In addition to the process include test that people from the client need.

# Scope

Inside the Scope

Once the test suites have been defined based on the requirements, 100% of the test cases (successful executions) will be used as an indicator of compliance for the following modules:

* Company
* Product
* Resources
* Customers
* Auditors
* Isgn In
* Get Started

Out of the Scope

Automated testing is currently out of the scope of the project, due to the maturity of the project, which will cost excessive time and effort to automate a page or application that is not yet in a stable version and is not yet developed.

# Quality Objectives

**Main Goal**

Comply with the modules established with the standards and contractual requirements, avoiding the technical indebtedness of each one the modules and/or the general application process. Prioritize usage, data, integrity, and transactions processed within the application, and prioritize customer requested modules.

These objectives will be covered by clarifying all the doubts with the client and in the sprints that are held, in addition to prioritizing the tests in modules that have a very high impact for the development of the system. In addition to not postponing many tests in the modules so as not to leave technical debt in any module.

**Secondary Objectives**

Identify and expose risks within the application, as well as communicate the status of the development to the team involved through a progress report based on the executions.

# Roles and responsibilities

Developer (DEV): Developer specialized in web development, responsibilities:

* Web development.
* Unit tests.
* Database connections and iterations.
* Publication of the versions of the application.

Product Owner / Business Analyze:

Intermediary that will indicate the needs of the client, creating requirements according to the needs of the business. The responsibilities of this resource are:

* Define criteria and specifications of user requirements.
* Report on the current state of the application to the client.
* Progress report based on the percentage of requirements developed and tested.

Quality assurance (QA):

QA specialized in UI / UX and manual tests, for the assurance of all requirements and specifications of the application. As such, your responsibilities will be:

* Creation of test cases.
* Execution of tests.
* Project progress report based on tests (metrics).
* Reporting and tracking bugs.
* Creation of test cases with automation if the client needs this part.

# Assumption for running tests

* For a change to be applied in the quality environment, it must comply with the unit test of the development team. These tests must contain the HAPPY PATH of the process being published. The test must be documented with evidence for the test to be considered effective.
* Regression tests will be performed each time a new version is released, and that version contains significant changes.
* Restrictions for the execution of tests.
* Once the smoke tests fail, the execution of the action is stopped, and the report will be raised to bring the correct flow.

# Testing Methodology

The purposes of the selected methodologies for the development and validation of Appointment Control will be shown:

* Prevention of defects in work products from the beginning.
* Bug tracking.
* Identification and prevention of risks.
* Visibility of development progress as problems.
* Definition of tests to be carried out and execution sequence.

Unit tests.

* Environment / Execution phase: Development
* Purpose: Review of new functions and / or modules before applying changes in the quality environment, to detect application failures by the developer and he can give the correct monitoring of the project. Once it is assured that the version is stable and does not present errors, the publication is carried out in the QA environment.
* This test is carried out by the developer, but this test is mentioned because QA can indirectly participate in this test by guiding the developer on business rules, techniques to test the application.

Integration testing.

* Environment / Execution phase: Development / Quality
* Purpose: Review of the integration of all modules.

This test marks the beginning of the review in the DEV / QA environment, in which the pretests of the application will be expected to work in integration with servers rather than interacting with interfaces.

Regression tests.

* Environment / Execution phase: Quality
* Purpose: Check that the application is not corrupted by new versions, either by last minute changes, new requirements, etc. This test can be applied every time a significant change is made to the project.

Re-test.

* Environment / Execution phase: Quality
* Purpose: Check that once a defect and / or changes are fixed to fix a bug in code to ensure that it has been fixed by the development team.

Smoke tests.

* Environment / Execution phase: Development / Quality / UAT
* Purpose: Valid when making a new publication, the application works normally.

Acceptance Tests.

* Environment / Execution phase: UAT
* Purpose: To verify by the business that the application works as indicated and expected. This test will define if the project is in an advanced stage of maturity for its next publication in a productive environment.
* In this test the participation of QA resources decreases, but they remain present to support the Product Owner in this stage.

# Suspension and resumption criteria

The test will be suspended of the following events are executed:

* If regression tests fail after a release, testing will be suspended until the root cause is found.
* At the time the application being reviewed in the quality environment, as well as in UAT, if most the scenarios fail, the information will indicate that the published changes are incomplete and that it will be necessary to validate that is missing of that it was interpreted wrong on the development side, to deliver a stable version with the right features.
* If the released versions have no releases errors and 100% regression/smoke test run is achieved, testing will resume as normal.
* IN the case of finding a critical defect, the execution will be stopped and the search for the root problem will be prioritized to avoid future problems based on the critical defect found.

# Deliverables of tests

As the part of the project tasting process, the following deliverables are generated from the QA side during the testing period:

* Test Plan: This document will contain all the information on how the test will be carried out, what types of tests, what methodologies, among other general information on how the tests will be carried out. This document is intended for development team stakeholders and quality resources to understand and follow.
* Test Suite: Set of project test cases that will be executed by the quality team.
* Bug Report Size: List of bugs found during the application development and their status. Project progress based on the execution and Graph of the percentage of test cases executed based on: Approved rejected, with error, pending and not applicable (metrics). A rejected test cases will be determined in such a way that it has to do with something that should not be tested or that based on what was tested, data is missing that is simply necessary to perform the test.
* Final test report: Document with a general summary of the executions carried out, defects found and status of the last execution.

# Metrics

* Test Progress: Total number of tests executed (successful) divided by the total number of test cases.
* Fixed Bugs: Total bugs fixed divided by the total bugs found.

# Defect report

The following data will be added to the project report:

* Priority: The importance of the defect being solved, which will be defined in three options:
  + High: The defect must be solved in the following versions due to the impact it has based on functionality or business logic.
  + Medium: The defect must be addressed after solving the high priority defects, so your publications may be postponed.
  + Low: The defect affects the application in a minimal way, so this defect can be delivered in the latest versions.
* Severity: The criticality and / or effect that the error has on the application, which is defined in three options:
  + High: The defect greatly affects the application or part of it, so the need to focus efforts on solving the defect should be prioritized.
  + Medium: The impact does not greatly affect the application, but its absence and the impact it causes on it is notable.
  + Low: The impact is minimal as the efforts to solve the defect.
* Description: General description about what causes the error.
* Steps: Steps to follow to get to the current error
* Expected result: Expected result without the current error.
* Current Result: Error triggered when following the indicated steps.
* Evidence: Evidence of the current error.