

AUTOMADB

**Test Plan and Strategy**

**Version 2.0**

Joaquin Miranda Castro

192493

AUTOMADBv2.0

[Version history 3](#_Toc40926760)

[Project information 3](#_Toc40926761)

[Approvals 3](#_Toc40926762)

[TEST PLAN 4](#_Toc40926763)

[Project Description 4](#_Toc40926764)

[Functional specifications 4](#_Toc40926765)

[Objectives 4](#_Toc40926766)

[Planification 5](#_Toc40926767)

[Testing Procedure 5](#_Toc40926768)

[Pass or Fail Criteria 5](#_Toc40926769)

[Risks 6](#_Toc40926770)

[Roles and Responsibilities 7](#_Toc40926771)

[TEST STRATEGY 8](#_Toc40926772)

[Test focus 8](#_Toc40926773)

[Resources 8](#_Toc40926774)

[Hardware 8](#_Toc40926775)

[Software 9](#_Toc40926776)

[Training 9](#_Toc40926777)

[Scope 10](#_Toc40926778)

[Tested elements 10](#_Toc40926779)

[Functionalities not to test 10](#_Toc40926780)

[Deliverables 11](#_Toc40926781)

## Version history

|  |  |
| --- | --- |
| **Date** | **Version** |
| 04/15/2020 | 1.0 |
| 05/11/2020 | 1.1 |
| 05/21/2020 | 2.0 |

## Project information

|  |  |
| --- | --- |
| **Organization** | SQA 2020 |
| **Project** | AutomADB |
| **Start Date** | 04/15/2020 |
| **Client** | Juan de Dios Delgado |

## Approvals

|  |  |  |  |
| --- | --- | --- | --- |
| **Team member** | **Date** | **Signature** | **Version** |
| Joaquin Miranda Castro | 05/11/2020 | Joaquin Miranda Castro | 1.1 |
| Ricardo German Serrano | 05/11/2020 | Ricardo German Serrano | 1.1 |
| Juan Carlos Patrón Ruano | 05/11/2020 | Juan Carlos Patrón R. | 1.1 |
| Joaquin Miranda Castro | 05/21/2020 | Joaquin Miranda Castro | 2.0 |

# TEST PLAN

## Project Description

AutomADB is a framework focused on automated tests for Android mobile devices, mainly used for testing phone applications that require human intervention.

The framework provides the tools for interacting with the apps in an automated way, and gives testers the way to write efficient, reusable test suites that are compatible with multiple Android versions.

AutomADB also offers compatibility with OpenSTF, an open source smartphone test farm, to be able to test remote devices with ease.

Overall, the main objective of this framework is reducing the time for automating test cases, improving the efficiency of the test development and allow the tester to focus his manual effort to more important issues.

### Functional specifications

* AutomADB must be able to connect to Android devices that are either connected locally to the adb server or remotely through an OpenSTF Device Farm. Either way, the framework should run its scripts if the device’s Android version is compatible.
* The framework must be able to run scripts that validate the following use cases:

Settings app:

1. Turn Wi-Fi on or off.

Calculator app operations:

1. Addition
2. Subtraction
3. Multiplication
4. Division

Phone app:

1. Call emergency numbers
2. Call national phone numbers
3. Call international phone numbers

### Objectives

* Test the functionality of each individual application’s processes.
* Develop an automated framework to accelerate the testing of these applications.
* Allow the easy inclusion of new test cases and test suites without affecting the functionality of the framework.
* Be able to easily support new android version without changing the test cases.
* Support OpenSTF for automated testing on remote devices using the local adb server.

## Planification

### Testing Procedure

During the development cycle of the previously stated applications an automated testing framework will be developed hand in hand to accelerate the testing process, thus offering a significantly greater amount of test coverage, more robust testing capabilities and overall efficiency. During the development cycle of the automated framework, visible issues will be corrected and will be logged adequately. After each build for the applications in test the automated framework will execute the test suite and a log with the test cases’ results will be recorded.

High priority defects or critical issues that offer no immediate solution will be raised and the development team will be notified, the test report summary will be updated for their resolution within the following sprint.

### Pass or Fail Criteria

The pass criteria for the automated framework is to adequately complete each test case within the target phone. For example, if the TC09 requires the phone to call a national phone number “77236019” the automated framework must complete this task on the tested mobile device within the phone application, and correctly store the result of the test case within the test log.

The automated framework is considered apt for delivery after 95% of the developed test cases for the framework are executed correctly. The libraries created within this framework must also include methods and classes which are reusable in case any of the applications under test requirements change. Thus, offering a significant advantage over manual testing.

Due to the fact that this framework was developed during a sprint the bugs or issues that are found after testing will be raised and will be corrected after further notice. All test results and bugs will be included in the Release Notes.

The pass or fail criteria will hold the same in either local or STF-remote devices.

### Risks

|  |  |  |
| --- | --- | --- |
| **Description** | **Probability** | **Mitigation** |
| OpenSTF does not officially support Windows. | High | Research cases where the Windows implementation was successful or create a Linux virtual machine. |
| Due to the ongoing global pandemic and the quarantine, access to Android devices with different versions for the STF farm is limited. | Medium | Focus on testing at least one device connected to STF. |
| Due to the fact that the phone app is the most critical part of a mobile device and is the feature that will most likely be used during an emergency by the end, the optimum functionality of this application is required otherwise a software bug could cause a life and death situation | High | Focus a great part of the automated testing framework on testing the phone app functionalities |

### Roles and Responsibilities

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Responsibilities** |
| Joaquin Miranda Castro | Developer | * Update the framework for the compatibility with STF. * Update the methods and libraries when necessary. |
| Joaquin Miranda Castro | SQA | * Update the documentation from the previous version. * Execute the test cases and report the results and bugs. |

### 

# TEST STRATEGY

## Test focus

The automated test framework is currently being developed to test mostly black box functional tests, utilizing techniques such as equivalence partitioning and boundary analysis, so we can give a good coverage to the test cases determined for the corresponding apps. The main testing tool being used are external libraries which are adb shell and UI Automator which allow the framework to access the mobile device and use its objects parameters to select them and utilize them as well as other activities such as swipe, type, search, long click etc.

The devices used for the testing will be remote devices connected to an instance of OpenSTF, which is an Android Device Farm. These devices will connect to the local adb server to run the scripts.

The testing is also based on the QA tester experience, error assumption and intuition to search for bugs or other issues that may arise in the automated framework.

## Resources

### Hardware

Hardware used to develop, test and run the automated framework:

Tested Phones:

* Motorola Moto G6
  + Serial: ZY323XP43F
  + Region: Bolivia
  + Android version: Android 9
  + Language: English

Hardware used to run STF and AutomADB:

* Acer Predator PH315-52
  + Processor: Intel Core i7-9750H CPU @ 2.60GHz
  + RAM 16 GB
  + Windows Operating System

OpenSTF suggest this hardware to run their service optimally: [https://github.com/openstf/stf#recommended-hardware](https://github.com/openstf/stf%23recommended-hardware)

### Software

Software y and its versions for the required automation framework

* Python 2.7 with Pip
* ADB shell
* UI Automator
* GitHub
* Any Python IDE.
* Java Development Kit 8
* OpenSTF

### Training

The participants of this project must research and comprehend Python, GitHub, UI Automator, ADB shell, OpenSTF as well as other external libraries that can be included within the automation framework.

## Scope

### Tested elements

1. Settings
   1. Wi-Fi functionality turn on and off, and validation of a successful connection.
2. Calculator
   1. Addition, subtraction, multiplication and division.
   2. Validation of a successful result.
3. Phone
   1. Call emergency numbers.
   2. Call national bolivian numbers.
   3. Call international numbers using the prefix ‘+’.
   4. Calls must be made via adb shell and / or UI automator.
4. Framework
   1. The Calculator should read its inputs from an external text file.
   2. The tests should be able to run on local devices or remote devices through OpenSTF.

### Functionalities not to test

1. Nonfunctional testing such as performance testing, security, usability, maintainability among others.

## Deliverables

|  |  |  |
| --- | --- | --- |
| **Elements** | **Description** | **Delivery date** |
| Test Plan and Strategy | A document which explains the ass and fail criteria, defines the individual test cases that will be developed, defines scope. | 05/22/2020 |
| AutomADB Demo | Partial product build which is then demonstrated to the stakeholders and hopefully given approval | 05/22/2020 |
| Test Cases and result execution | Document that is developed to thoroughly test the automated framework, finds bugs and issues during execution. Results are then inputted. | 05/22/2020 |
| Release notes | Document which includes the results of the test suite execution and informs the team if the product build is ready to enter the next stage of development, and what issues or bugs must be resolved during the following sprint if any are encountered. | 05/22/2020 |
| Business Case | Document which is presented to the stakeholders to inform them in the most complete form what our automation framework is, its goals, benefits, ROI, architecture, Level of Effort, etc. | 05/22/2020 |
| Traceability Matrix | Matrix which joins the user stories and test cases that were executed based on the project’s requirements for the product owner to see. | 05/22/2020 |
| OpenSTF integration | The newest framework feature, the capability to connect to remote devices from an STF instance and be able to run the test suites on them. | 05/22/220 |