AUREOS: Testing

Software Validation

B. Ruiz Sabido Bryan F., B. Lizarraga Franco Mauro J., B. De la Cruz Ramos Carlos J. & B. Martinez Contreras Yeshua Javier. Universidad Politécnica de Yucatán, Tablaje Catastral 4448, Carretera Mérida-Tetiz. Km.4.5, 97357 Ucú, Yuc. Embedded Systems Engineering. Advanced Programming. November, 2018.

INDEX

I. Test cases & Validation techniques

● "Validation: Are we building the right product?"

● "Verification: Are we building the product right?"

Development testing

1.1. Unit Testing

- Enter to the App.

Test Case #1. Check if the music sounds.

Test Case #2. Check if the targets display the image.

Test Case #3. Answer a message and the return to the App.

Test Case #4. Answer a message while we use the App.

Test Case #5. Display 3 targets

Test Case #6 Change quickly all the targets.

1.2. Component testing

Answer a message and return whereas we display all the targets.

Enter to the App and Check if the music sounds.

II. Software Quality Testing

First Method: Black-Box

Focusing primordially in the external behavior, ‘Black-Box’ allows to test any AR application, slightly efficient. In spite of focusing in how users get in touch with the functionality, the perspective should be, in how the environment interferes in such of convenient situations where Aureos take approach. Black-Box attempts to find errors in the following categories (Sommerville I.):

1. Incorrect or missing functions
2. Interface errors
3. Errors in data structures or external database access
4. Behavior or performance errors
5. Initialization and termination errors

Taking this into account, Black-Box take a new vision in how the user interacts with his normal paradigm of seeing the world that surrounds him; this translates to Aureos, for having an actual concentration in the camera movements, targets recognition and the database link that is being stimulated during the running of the app.

The regular sets of inputs and test cases that were mentioned before, staged on the Black-Box technique, making the delimitation of objectives, almost digestive; nonetheless, Black-Box only takes the relationship between the user and the external factors, which means that other scenarios are not taking into account. To solve this situation, another technique of validation is taken.

2.2. OBJECTIVES WITH BLACK BOX TESTING

First of all, we need that the application opens it up without no interruption or forced close when running application, we need this testing because it is very annoyed for any user when an application closes, most of the people uninstalled application because their cellphones don't run correctly that application and that will be critical for any company or entrepreneur. In play store there are many applications that are not compatible with many cellphones and the most of those applications are at the bottom of the rating.

As second objective we need that the application can recognize every target and doesn't' get stuck when changing a target and also all in all, the application must run all the targets, if a target doesn't work, our programming team will change the function of that target in order to repair it. It's very important that the target works because our main goal is to replace paper schedule and name of the classroom, which are pasted on the door of any classroom, by an only target.

So, we thought about the next tests:

1. **To Give them the link to download to see if it is easy to do.**

2. **To Check how well they used the App.**

3. **To see if the location of the targets is correct.**

2.3. Second Method: Monkey Testing

As a matter of fact, XR apps use the graphic engine of mobile devices. This demand of resources of the system, can set up tons of events during the day life of the user. Monkey Testing will allow to test Aureos in almost every single scenario where the user and the app, get in touch. It performs the regular routines of an application; how many crashes can occur through different platforms and/or devices.

For allowing the use of Monkey Testing, terms are defined below:

- A number of testers will be defined
- Random devices should be reported
- The support to iOS and Android is mentioned in the system requirements, which means that any device with a lack of those, will not enter into the test.
- The routines that every user make of their apps will not contrast with the test cases.

2.4. Objectives with Monkey Testing

1. The App is tested on the night and with low brightness.
2. All the targets together.
3. Under a raining.
4. Use as many cellphones.
5. Put all the targets then put the App in second plane and then return to the App.

III. ERROR ANALYSIS

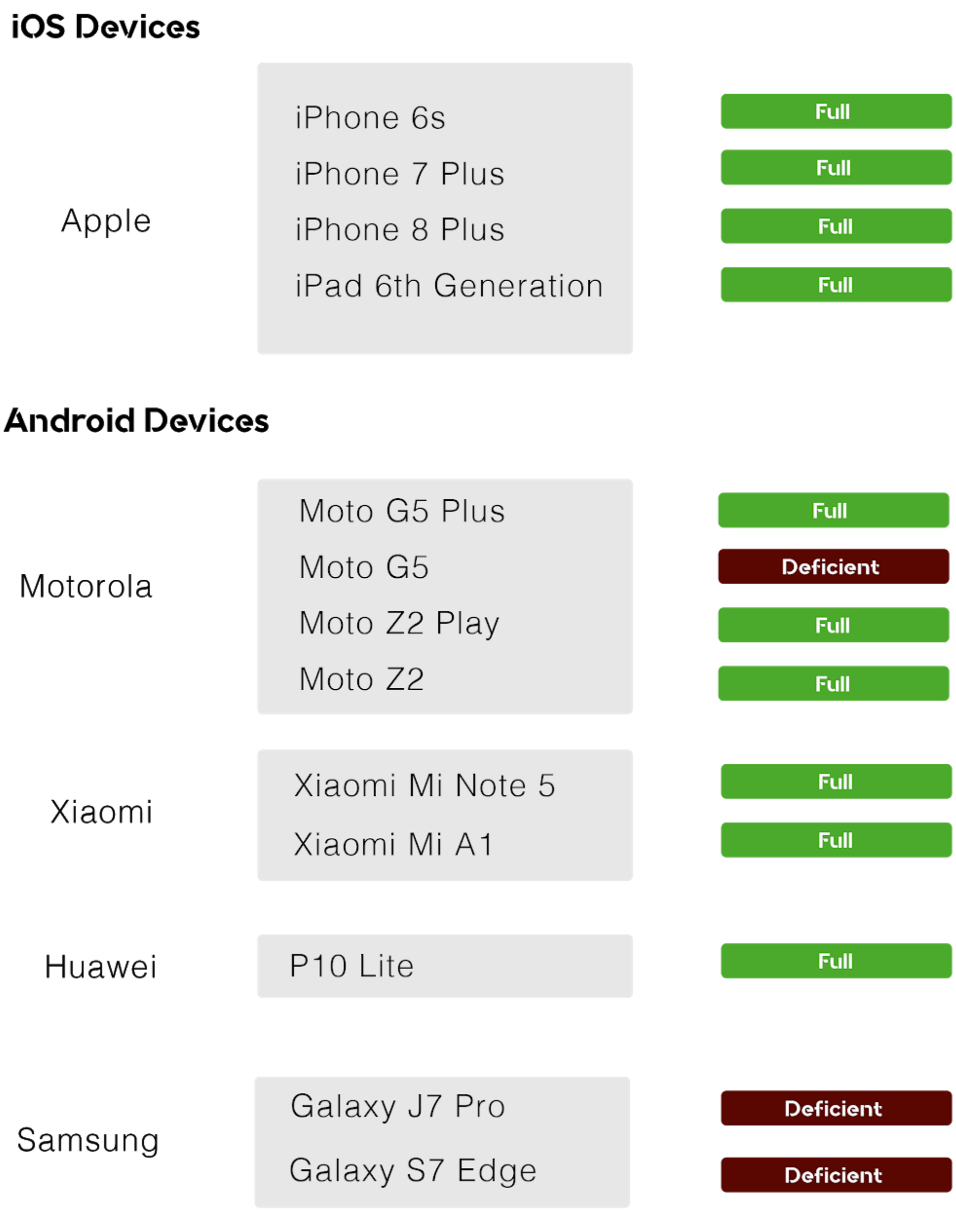
3.1. REQUIREMENTS MATRIX

REQUIREMENT	Functional design	Requirement description	DESCRIPTION
B1: Camera and sound permissions given by the user.	Required	When opening the application, it must display an advertisement window to the user asking for permissions to allow the app to use the camera on the phone.	**Customer:** For using the app the customer must agree with the use of the camera and sound of the mobile.
B2: Vuforia Database	Required	All the targets must be stored on vuforia's database.	**Owner:** The owner must have control of the database in order to change a target or the image displayed.
B3: Crossplatform	Required	Aureos must run on both iOS and android operating systems.	**Customer:** The application must work on both systems and it must have the same behavior on both OS.

B4: Support for iOS since 9.0, and iPhone 6s and above.	Required	For iOS operating system, every iPhone must have iOS 9.0 or above.	**Customer:** For using the application, customers with iOS devices must have iOS 9.0 or above with 2GB RAM (since 6S).
B5: Android version since 4.1 and devices with at least 2GB of RAM (for recommended latency stability).	Required	For Android systems, Aureos application must run efficiently.	**Customer:** For using the application any customer with Android system must have an android version 4.1 or above and with 2 GB of RAM.
B6: No logging system to enter the app is required.	Required	Application doesn’t need an account to access.	**Customer:** No customer needs to create an account or login with any kind of account. **Manager:** No account is needed.
B7: Display proper information from the target.	Function	The target must display the proper file when focusing the camera on it.	**Customer:** When the customer focuses the cellphone, the camera must display the correct image for the target in focus.

B8: Aureos Logo Overview	Required	After taping on the application icon, the Aureos logo must be displayed once.	**Customer:** Any customer must see the Aureos logo after tap on the icon application.
B9: Display MULTIPLE targets	Function	When focusing Cellphone camera with multiple targets, the camera must display each target at the same time without interference.	**Customer:** If the customer wants to watch two or more targets at the same time, the application must display the correct image for any target when it is focused.

3.2. ERROR ANALYSIS REPORT



iOS Devices

Apple	iPhone 6s	Full
	iPhone 7 Plus	Full
	iPhone 8 Plus	Full
	iPad 6th Generation	Full

Android Devices

Motorola	Moto G5 Plus	Full
	Moto G5	Deficient
	Moto Z2 Play	Full
	Moto Z2	Full
Xiaomi	Xiaomi Mi Note 5	Full
	Xiaomi Mi A1	Full
Huawei	P10 Lite	Full
Samsung	Galaxy J7 Pro	Deficient
	Galaxy S7 Edge	Deficient

- *Apple devices were running iOS 12 and above.*
- *All Android devices were running Nougat, except on Galaxy S7 Edge.*

3.3. TEST CASES REPORT

Test Cases Description

- Test Case #1. Check if the music sounds when opening Aureos.
- Test Case #2. Check if the targets display the overview.
- Test Case #3. Attend a notification, then, return to the App.
- Test Case #4. Answer a message from notification while using the App.
- Test Case #5. Display 3 targets.
- Test Case #6. Change quickly the target that is being scanned.
- Test Case #7. The App is tested on the night and with low light.
- Test Case #8. Under the rain.
- Test Case #9. Scan all the targets then put the App in second plane and then return to the App.

Statistic Report

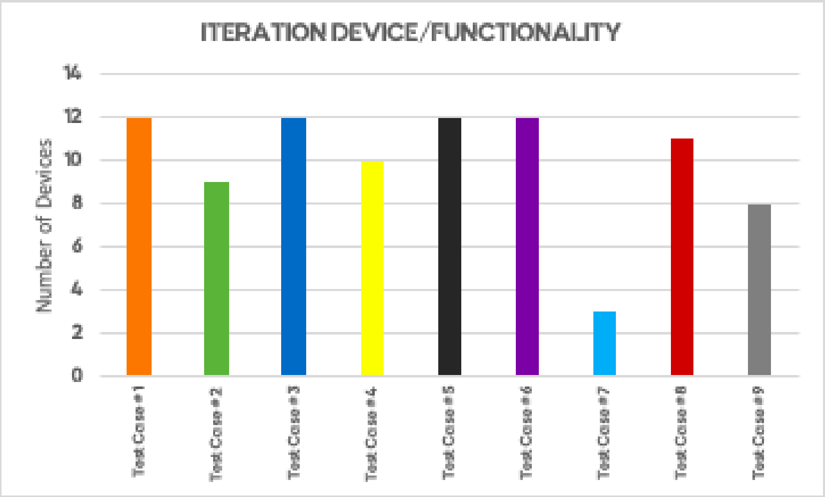
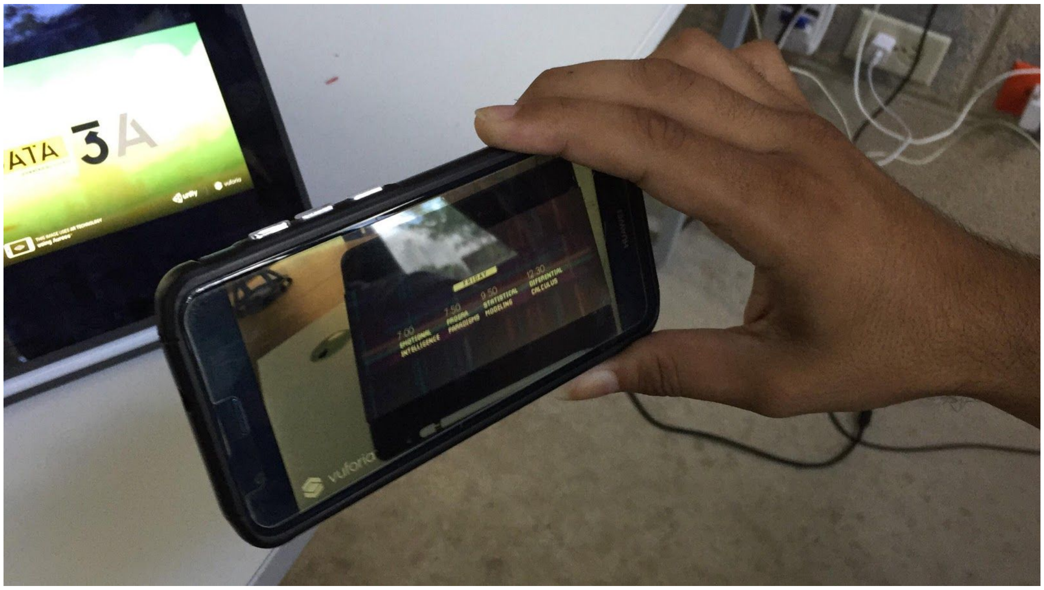
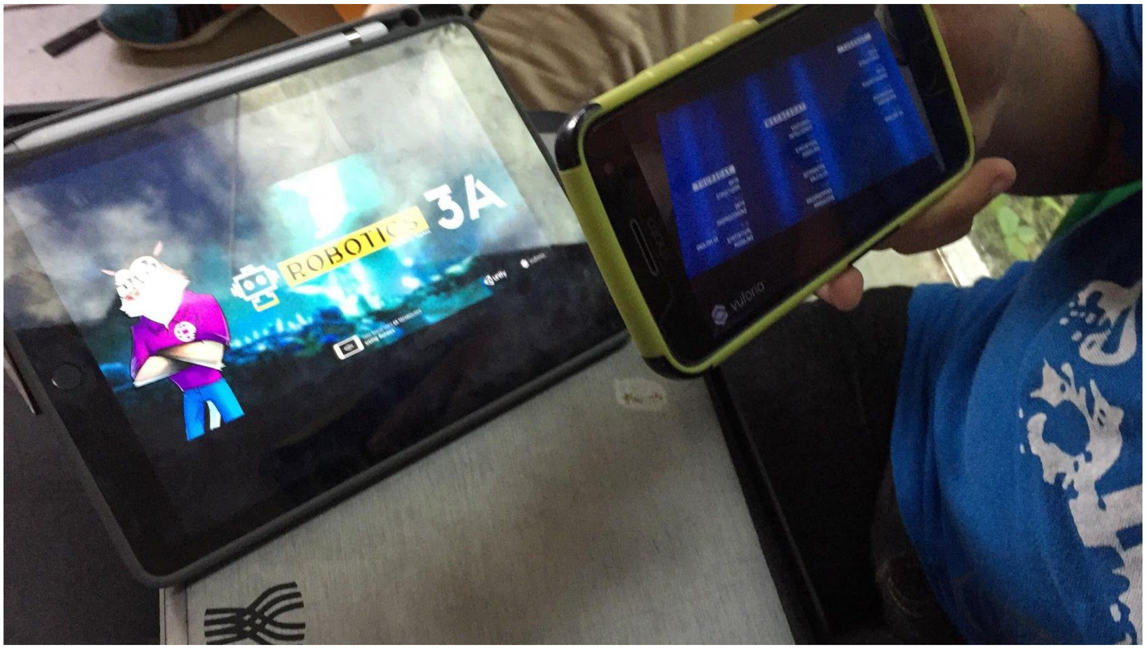


Chart 1. Iteration Device/Functionality

As it was shown in the chart above, the amount of devices that succeed in the test case previously defined was quite high. It should be mentioned that there were 4 out of 9 test cases where every device ran as it was expected. Nevertheless, the Test Case #7 was the case with lower success rate by testing Aureos functionality on low light conditions.

Photographic Memory


ATA
3A
unity
vuforia
FRIDAY
7:00 EMOTIONAL INTELLIGENCE
7:50 PROGRA PARADIGMS
9:50 STATISTICAL MODELING
12:30 DIFERENTIAL CALCULUS
vuforia



Check the whole testing on:

https://www.youtube.com/channel/UCsqeP5U6bQy_ZgeUp5QtbCQ/featured?view_as=subscriber

Events Binnacle

Monday, November 5, 2018

The design team was figuring out a way to perform the same functionality through cloud storage. Using Dropbox, we did some of the first developer tests.

Through a cloud storage service, in this case, Dropbox, the videos that targets should display are stored in the cloud and not in the app. This way we can make the app weigh less since the displayed videos are streamed from the cloud and not from the app itself.

Friday, November 9, 2018

On iOS devices, we tested many devices and the lower and higher devices were iPhone 8 Plus and iPhone 6S. The rest of iOS devices between the mentioned before were also tested during the same lapse. This gave us an idea of the devices that are in the edge of testing. That means Aureos won’t work on devices before iPhone 6S and it will probably have some issues on devices above iPhone 8 Plus.

Saturday, November 10, 2018

Although the team developer had set up the different test cases that can be applied through our methodology, they had not been written in the document, yet.

Wednesday, November 13, 2018

We had the opportunity with some new devices that we never tested before, those are Motorola and Huawei. Besides, the rainy day that the team faced, made possible to run out the Test Case #8. Data was gathered and some photos were taken as evidence.

3.4. VALIDATION STANDARD AND EMPLOYEES PERFORMANCE

IEEE Std. 1061

Identification of software quality metrics

Implementation of software quality metrics

Analysis of the results of software metrics

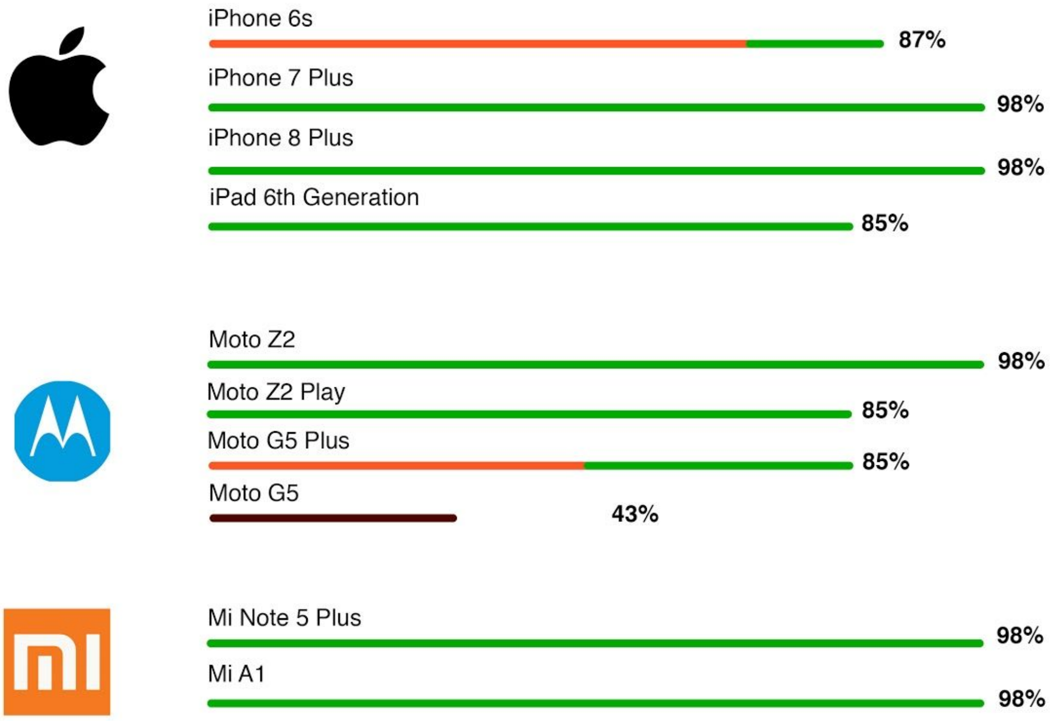
Validation of software quality metrics

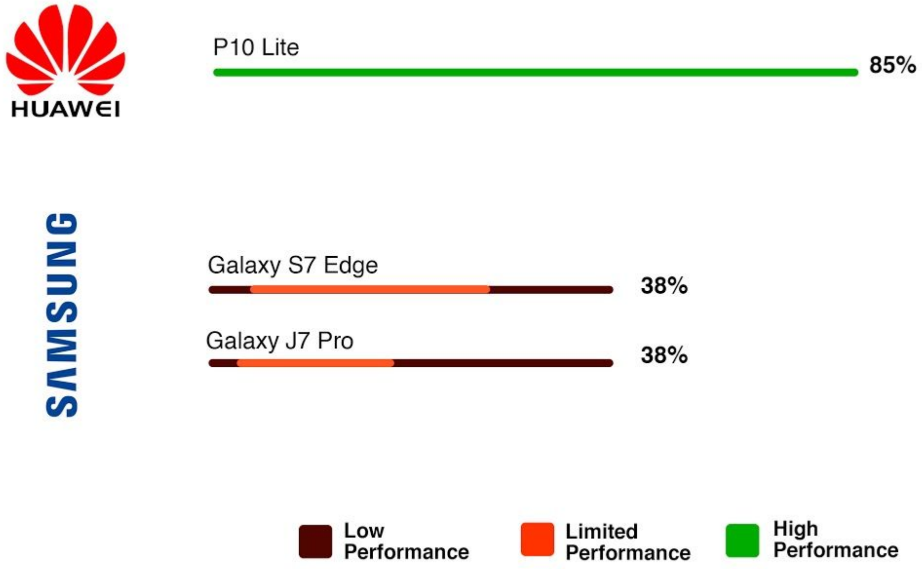
Methodology

- Scope of quality goals
- Establish quality requirements
- Establish Criteria
- Evaluate quality level
- Detect Anomalies
- Monitor Changes
- Validate system control
- Purpose of the Standard Protocol
- IEEE Std 1061

3.5. ERROR STATISTIC REPORT

The level of support in devices, either fully functional, partially functional or nonfunctional, collected a considerable amount of data for the system requirements that we can use. As a matter of fact, the iOS devices performed and passed run tests with no issue. On the other hand, Android devices were quite stable in the majority of smartphones. Nevertheless, Samsung phones did an non efficient performance. The results are shown below:

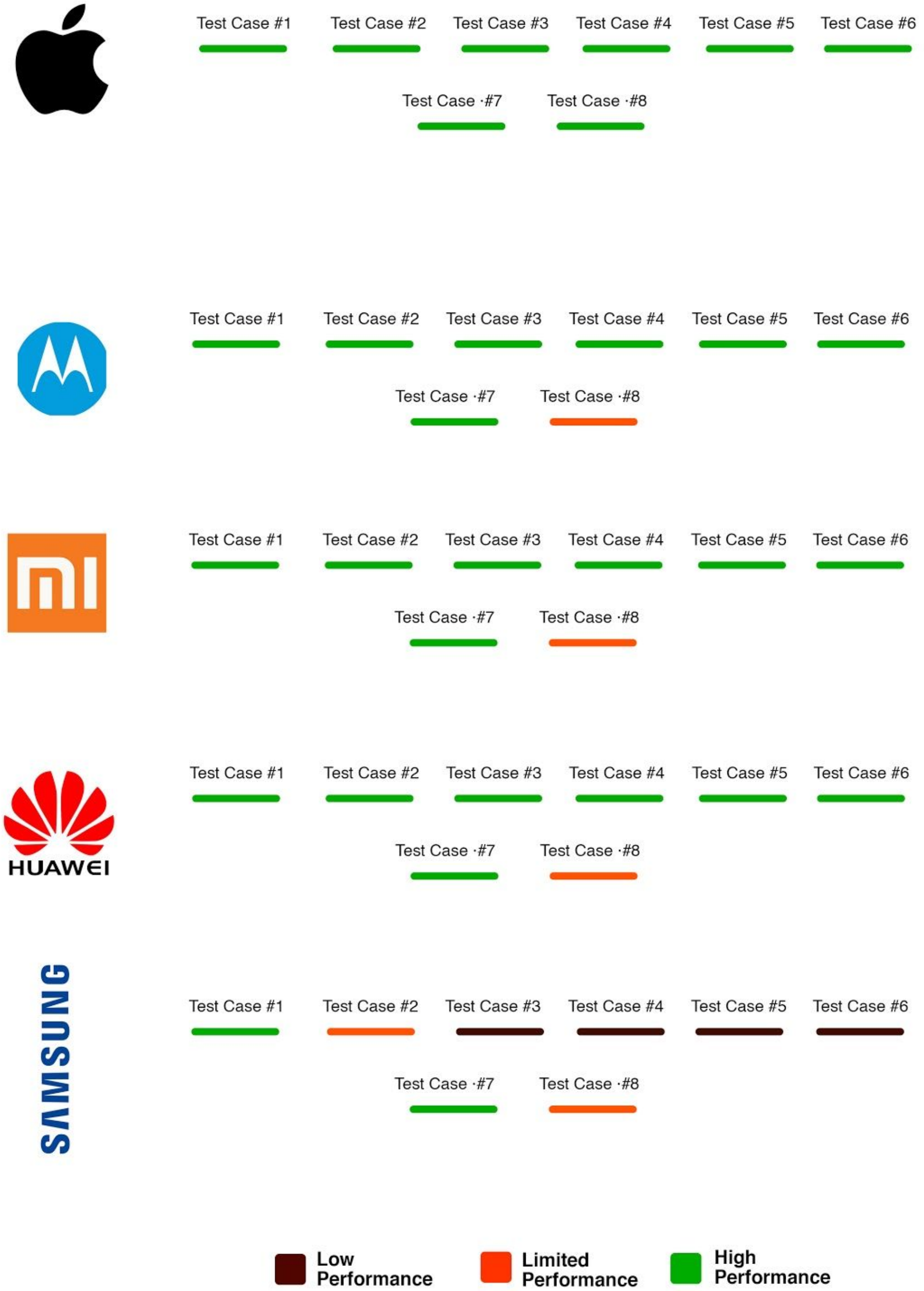




3.6. TEST RESULTS ANALYSIS

Based on the tests done in several brands of phones, we got the followings percentages: Xiaomi (95% - 98%), iOS (85% - 98%). On Motorola G5 Plus, Moto Z2 Play and Moto Z2: The results were the expected, the targets were displayed so quickly and talking about the part of testing they behaved quite well, and the only different aspect were the specs of the phones, opening the app and returning to it was where the MotoZ 2 had the advantage on speed.

Samsung j7 pro, s7 edge, (and other samsung devices): Here the behavior of the each phone was the same, taking into account devices from the lower gama to the highest one the results were the same, the targets were not displayed by the Aureos app.



Test Case #1 🟩
Test Case #2 🟩
Test Case #3 🟩
Test Case #4 🟩
Test Case #5 🟩
Test Case #6 🟩
Test Case ·#7 🟩
Test Case ·#8 🟩
Test Case #1 🟩
Test Case #2 🟩
Test Case #3 🟩
Test Case #4 🟩
Test Case #5 🟩
Test Case #6 🟩
Test Case ·#7 🟩
Test Case ·#8 🟧
Test Case #1 🟩
Test Case #2 🟩
Test Case #3 🟩
Test Case #4 🟩
Test Case #5 🟩
Test Case #6 🟩
Test Case ·#7 🟩
Test Case ·#8 🟧
HUAWEI
Test Case #1 🟩
Test Case #2 🟩
Test Case #3 🟩
Test Case #4 🟩
Test Case #5 🟩
Test Case #6 🟩
Test Case ·#7 🟩
Test Case ·#8 🟧
SAMSUNG
Test Case #1 🟩
Test Case #2 🟧
Test Case #3 🟫
Test Case #4 🟫
Test Case #5 🟫
Test Case #6 🟫
Test Case ·#7 🟩
Test Case ·#8 🟧
🟫 Low Performance
🟧 Limited Performance
🟩 High Performance