Alexa, How to test Alexa?
An Automation approach

Author: Juan Delgado

Co-Author: Lucaz Tutur, Piotr Wroblewski, Deanna Raven, Christopher Schoppa

juan.delgado@mobica.com, lucasz.tutur@mobica.com , piotr.wroblewski@mobica.com, christopher.schoppa@mobica.com , deanna.raven@mobica,

Abstract

This paper presents an idea or a concept for test automation on gadgets to use voice assistance.

Nowadays the Voice assistants, like Alexa are taking over households, in the USA anyway, where one in four households owns at least one device. Those gadgets provide an increasing part of the added value of modern gadget systems and thus become more complex to test.

Where one of the most common general problems with the test automation framework working with hardware and software to test the voice assistance seems to be forgetting that any test automation project is a software project in its own right. Software projects fail if they do not follow processes and are not managed adequately, and test automation framework projects are not different. Of all people, test engineers ought to realize how important it is to have a disciplined approach to software development.

This paper focuses on developing a Proof Of Concept to approach an automation framework running a script on Python and using different libraries that strictly follow the QA best practices and coding to execute a common E2E Test Scenario, where the main goal is to approach a quality strategy to test the Alexa voice assistant on real android gadgets to validate "Hey Alexa" in different acoustic noise conditions and different utterances to validate how many times is successfully detected; looking to help on QA community sharing knowledge to foster a software testing quality.

Biography Authors

Juan Delgado

Juan de Dios Delgado has more than 15 years of experience as an engineer. Currently as Test Manager, with Mobica. With strong project management skills, key areas of expertise include understanding complex business requirements & formulating robust test strategies; developing automated test solutions with the ability to interface between Development, Project, QA & Test Teams to ensure execution of test strategy; and extensive software engineering skills.

He has participated in several projects, in the last years. He has worked on-site in the US and Europe; near shore from Mexico to the US, and offshore from Mexico and India to Europe; working strongly with Functional Testing focusing on Automated Testing for Mobile devices, Web Applications, and Web services.

Professor/Lecturer at Universidad Tecnológica de Aguascalientes and Universidad Panamericana in Aguascalientes, México.

Lucasz Tutur

Lukasz Tutur has over ten years of experience in various projects for top companies. Firstly, system tests of Broad Band Access OSS systems, running both in Windows and Unix environments where his responsibilities included LSP and functional testing, using black/grey box techniques, incident reporting, test specification writing, and test list designing.

Co-authors:

Piotr Wroblewski

Head of Delivery Mobica US. 20+ years of experience in creating, defining, developing, and delivering innovative global telecommunications, defense, cloud, and software technology products, services, and solutions.

Christopher Schoppa

Chris is Head of Solutions for Mobica USA. He has spent his career at the intersection of technology and business value across large volume web applications and services, automotive infotainment/digital cockpit, mobile devices, and IoT. I work to understand your product strategy, then come up with ways to use appropriate technology and deliver novel use cases that grow your business.

Deanna Raven

Senior Program Manager/Senior Technical Project Manager with 12+ Years' experience across the software industry with a passion for leading change efficiently and an unwavering enthusiasm for solving business problems and uncovering new opportunities using the power of technology and data.

1 Introduction

We are living in an age of smart devices, where voice assistant applications or devices have become the access point for controlling several functions daily. Some of these popular devices are Alexa by Amazon, Google Home, and Siri. Besides most new smart gadgets that are launched into the market nowadays also have inbuilt voice assistant features.

For example, Alexa is one of the most intelligent voice assistants that can solve diverse queries on specific gadgets which is a must need to do functional testing on the voice assistance under the most popular gadgets thoroughly in order can deliver an optimal user experience.

Test the voice assistance feature of the devices may seem simple on the surface, but in fact, many challenges determine the success of the skill

This paper is willing to show a proposed proof of concept with the quality approach testing the Alexa voice assistance under some gadgets as well as the technologies and methodologies implemented on the strategy besides the results and benefits.

2 Voice Assistance

Voice Assistant is a virtual assistant that uses speech recognition, natural language processing, and speech synthesis to take action on user utterances to help in our day-to-day and they are available in cars, household devices, smartphones, and several apps.

As mentioned, this paper will discuss how we can get into voice assistant testing, and we will propose a QA approach to develop a black-box automated functional test Proof Of Concept to test a voice assistant, specifically on Amazon wearables, then Hey Alexa! please let's start.

2.1 Background information

Much like the structured language used by humans, where a word signifies an object, digital assistants rely on a combination of utterances and intents to respond to voice commands.

- Utterances: An utterance is an input from the end user. It may be a sentence, such as “play baby shark on amazon music,” or a fragment of a sentence, such as “open amazon music”. Utterances are impacted by variations in user location, gender, age, etc.
- Intent: The intent is a purpose or goal expressed in a user’s input (utterance), such as playing baby shark on amazon music. An intent represents what the user wants the digital assistant to do.
- Entity: An entity represents detailed information that is relevant to the utterance. For example, in the utterance “play baby shark on amazon music on Fire TV” where Fire TV is an entity.

2.2 Alexa

It all starts with Amazon's wake word "Alexa", this voice assistant is part of the Amazon services that are built around the Alexa devices to support voice commands. Alexa voice service was first introduced with the Echo device. To interact with a smart device, a user must say Amazon's wake word “Alexa” and then say a certain command. The device sends then the user commands to the Alexa voice service which is the cloud service responsible for processing the voice commands and returning the response back.

On a generic workflow explained, a user can say Amazon’s wake word “Alexa” plus a command (called "utterance") into a device, This utterance inputs are interpreted as voice commands that are detected and

recognized. The actual output can be seen as a voice answer given by the device used to the user, or also can be a performer action on the device; as a Software developer in Test even could be an output value called "logs" shown on the Logcat window in Android Studio.

3 The Voice Assistance testing – the big picture

Voice assistance presents a huge opportunity for companies could engage with their users, however, such as is complex technology increases the difficulty of testing for that reason a correct quality strategy approach is essential for the best user engagement without negative impacts on their experience.

Common challenges for testing voice assistance include:

- User: A wide range of variables with different languages, accents, ages, and gender.
- Device: Lab with required devices as 1st party (Amazon Echo, Google home, etc.) and 3rd party (android devices, cars, smart gadgets, etc.)
- Environment: Different background noises.

3.1 Proposal

In this paper, we will focus on black box testing to test the trigger of Amazon's wake word "Alexa" and to validate the performing actions with an in-house build python script besides using different tools and methodologies that support the most common aspects of quality approach in the proof of concept that will be shown using Amazon wearables that explore a real-life example to help you get started with voice app testing.

3.2 Goal

The QA approach strategy consists of building a test automation script that plays an utterance using Amazon's wake word "Alexa" on a computer or external speaker to be triggered on wearable Echo5 and performed on Android Mobile Phone or Fire TV apps.

Our potential solution is to define an approach based on an Automated test solution using python to perform the speech of Alexa and to read the Android logs in the wearable to validate if the trigger was successfully accurate and the action performed and willing to be shown in a demo as proof of concept.

4 Project Methodology

Let's look at some of the methodologies applied in our Proof of concept to achieve a successful demo.

4.1 Scrum

Working with Scrum methodology begins with identifying the problem and its core components, in this case, the team focused on the MVP. Knowing this we work in a series of sprints to achieve our goal.

The team was divided into three roles that ensure the workflow needed. First, we have the role of Scrum Master, this role is akin to a coach, who helps the team with its expertise. He focuses on improving the team's effectiveness.

The next role is the Development team, they work to deliver a potentially releasable increment of finished tasks at the end of every sprint. Their task is to give structure to the development project and determine the number of tasks given to each member. They also carry out the task of working on writing the paper.

Finally, we have the role of the Software Developer Engineer in Test, this person writes the tests proposed by the Development team, using all the given tools, for identifying the possible flaws of the web platform. This person analyses the outcome of each test and documents their conclusions. Since this is an automated framework, the Developer also integrates the tools needed for the scalability and automation of the Proof Of Concept.

4.2 MVP

To define our proof of concept and build a demo to be shown to the PNSQC 2022 audience, we based our project on the true meaning of a minimum viable product (MVP) focusing on our python script with the Minimum set of features of being viable.

We based our solution on the following:

- Execute the speech of the Hey Alexa
- Validate the triggering of Amazon's wake word "Alexa"
- Validate the response of the triggering
- Validate action performed in the wearables.

4.3 Best practice testing

4.3.1 Testing

During the development of this automated framework, our team focused primarily on three main types of testing. Functional testing, black-box testing, and regression testing.

4.3.2 Testing techniques

To improve the quality of the tests there are various techniques during the design process of a test suite. Our design process was dynamic, a combination of specification based with experience-based techniques.

The specification-based techniques are primarily based on black-box testing which included techniques of its own such as use case, user story testing, state transition testing, boundary value analysis, and equivalence partitioning, among others.

Using experience-based techniques, such as exploratory testing or error guessing also is important to be considered in the test design process due the team might be composed of engineers with different skill sets and backgrounds and this could allow having a variety of perspectives and insights to lead a better robust test coverage.

For our Proof Of Concept and following the MVP, Scrum methodologies, and testing techniques, our testing will cover the following functional test cases that will be shown in our demo presentation.

- **Hey Alexa – Play a song!**
  - Functionality: When the Amazon Echo is connected to a Fire TV Amazon's digital assistant, Alexa, can help to play movies and TV shows from Amazon Prime Video, Hulu, and more using the wearable Fire TV. Also, can do most of the common functionalities such as stop, play, etc. with voice commands.
  - User story: As a Software Developer Engineer in Test, I need to validate that when Alexa wake-word is triggered should be detected by the Amazon Echo 3 to perform the utterance action on Fire TV.

- **Hey Alexa – Make a phone call!**
  - Functionality: When the Amazon Echo is connected to an Android device Amazon's digital assistant, Alexa, can help to make a phone call using the wearable mobile phone.
  - User story: As a Software Developer Engineer in Test, I need to validate that when Alexa wake-word is triggered should be detected by the Amazon Echo 3 to perform the phone call action from the mobile phone.

4.4 Best practice coding

There's no standard process when it comes to coding. Therefore, industries have been working on creating rules or tools that ensure that all code has the same structure and has an organic flow throughout the code structure.

Below we discuss a few best practices and techniques that we have used in the development of our demo to enhance the coding of the Proof Of Concept.

4.4.1 ESLint

ESLint is a tool that finds and fixes structural and compilation problems in the code. ESLint is a pluggable tool that helps identify and report patterns found in JavaScript code, with the goal to make the code more consistent. You can make your own set of rules, or you use the rules that industries like Google or Airbnb employ.

4.4.2 Code reviews

Code Review is the process in which a programmer consciously and systematically checks the code for mistakes or structural mistakes. The code reviews have proven that it helps to accelerate the streamlining of the development process for projects.

Code Reviews, when done right, can improve the programmer workflow, reducing the amount of time the Quality Assurance Team requires to check the code, therefore saving time in general.

Particularly in this project we employed the Over-the-Shoulder style of code review. We decided to use this technique because it's the easiest and more intuitive type of style to adopt. Once the code is ready, the supervisor of the code downloads the branch of GitHub that has the new code that needs to be reviewed. Once the code is reviewed and the team agrees with the changes, the programmer makes a merge from the review branch to the main branch of GitHub.

5 Framework

The voice assistance undergoes rigorous functional tests that ensure the correct functionality of Amazon's wake word "Alexa" and its performance. Usually, we have two types of testing: manual and Automation testing.

By implementing the appropriate framework for automated testing, we can significantly increase the speed and accuracy of testing, providing a higher ROI from the project.

Following the standards for a good Framework would achieve relevant automated tests, concise reporting, team consistency, implementation, and maximizing re-usability on the automation testing.

5.1 Tool

We will now talk about the main tools used in a particular way:

5.1.1 PyCharm

PyCharm is the most popular IDE for Python and includes great features such as excellent code completion and inspection with an advanced debugger and support for web programming and various frameworks.

We will use PyCharm for the Test Automation project for running tests automatically, managing test data, and taking the results to improve the quality approach.

5.1.2 Android Studio

Logs are very useful when Software Developer Engineer in Test is diagnosing an error or validating an expected value output from the Android tested, which includes several logs that deal with different parts of our performance of the testing,

The Logcat window in Android Studio displays those Android logs in real time and helps us to use it to collect and visualize those logs.

5.1.3 Audacity

Audacity will help us with the different audio utterances on m4a file format that need to be recorded, edited, mixed, and added effects to be played during the test execution according to the test case.

5.1.4 GitHub

A tool that as a programmer, the more you use the more you begin to depend on it due to its incredible ability for version control and management. It allows us to collaborate remotely with all team members, something that today has even more value due to the ongoing world situation. This makes it perfectly suited for our project. Thanks to our use of the agile Scrum methodology we will have a higher quality and security of deliverables for each sprint, which is combined with a software architect who will constantly help us have better code in our project.

5.2 Core (code structure)

5.2.1 Suites

This section contains the Tests files. Each file contains the tests that were identified as inherent for assessing the correct functionality of Amazon's wake word "Alexa" triggered.

- **Alexa-youtube.py:** This contains the following tests - Alexa plays a YouTube video on an Android device.
- **Alexa-phone-call.py:** Contains the following tests: Alexa makes a call on an Android device.
- **Alexa-firetv.py:** Alexa plays a video on Fire TV

5.3 Libs

5.3.1 Adb Folder

Run an ADB command and other functions that will execute specific commands like install apps, open apps, close apps, make a phone call, download logcat, etc.

5.3.2 My App Folder

Create an object with the package name of the app and perform a specific action in the app.

5.3.3 My Device Folder

Create an object with the ID of the device with main functionalities to perform

5.3.4 Utils Folder

Contain the functions to help us to play an utterance on the computer.

6 Conclusion

6.1 Outcome

This project is just a glimpse of how to start an approach of quality on voice assistance functionalities that might evolve throughout the Software Testing Life cycle to create an Automated Framework.

Mobica being part of the PNSQC 2022 thinks that can contribute to the QA community by sharing experience and best practices so the community interested can start their own automation project.

References

https://www.einfochips.com/blog/addressing-the-common-challenges-in-voice-first-testing-through-automation/

https://www.codexa.net/alexa-skill-kit-beginner-tutorial/

https://techwiser.com/fix-youtube-not-working-amazon-fire-tv-stick/

https://www.slideshare.net/ApplauseMarketing/solving-the-top-problems-of-voice-testing?from_action=save

http://www.diva-portal.se/smash/get/diva2:1473113/FULLTEXT01.pdf

https://alan.app/blog/voiceassistant-2/

https://www.slanglabs.in/voice-technology-glossary/voice-assistant

https://www.softwaretestinghelp.com/software-testing-techniques-2/#User_Story_Testing_Agile

https://www.audacityteam.org/

https://www.jetbrains.com/

https://developer.android.com/studio