**Challenge Three:**

**Description:**

As a QA Engineer I want set up a draft / concept for a pipeline that covers CI/CD of an iOS and Android mobile App project that includes a big UI testing suite in a nightly fashion and a quick smoke check.

**Acceptance Criteria 1:**

Which test engine / tool do you want to use and why?

**Proposed tool and reason:**

Appium is the impeccable tool to test iOS and Android Mobile App . Though there are many mobile automation tools available at present, it proves to be the most effective and accessible.

Appium is flexible, feature-rich, scalable, and hugely helpful. And learning it would ease the job of every mobile tester and it is just an open source mobile application for mobile automation. It supports native, hybrid, and web application automation test across physical devices including both an emulator and stimulator.

Appium let the user write tests for multiple platforms including iOS, Android, and Windows Phone using the same API. It means that you could write a test for iOS and use the same test for Windows Phone and Android.

Another great feature of Appium is that it has no dependency on Mobile device OS. It is because it has a framework or wrapper that could translate Selenium Webdriver commands to UI Automation or UI Automator commands based on the type of device. Furthermore, it supports almost all languages which have Selenium client libraries.

Appium even showcases a neat feature that can automate without recompiling the app. It means that the user doesn’t need to have access to the application code to work with it. Whereas, other mobile automation tools like Frank and Calabash demands the user to be aware of the app code to be able to work with it.

**Conclusion:**

So Appium is the right choice to test the iOS and Android mobile application testing.

**Acceptance Criteria 2:**

Simulators vs Real Devices? Elaborate the Pros and Cons.

**Simulators and Real devices Pros and Cons:**

When it comes to simulators/ emulators and real devices, each solution has its benefits and drawbacks, depending on the stage of an app’s lifecycle. Simulators/ emulators are great for early stages, but testing on real devices before releases is crucial to ensure an app’s success. “A recommended approach is to find a healthy mix of emulators, simulators and real devices to get the best out of your test automation.”

**Simulator Pros:**

* Simulator sets up a similar safe environment to the original real-life device’s OS.
* Fast (because they simulate only the software)
* Relatively easy to set up
* Can be used to study the behaviour of an app

**Simulator Cons:**

* It doesn’t take into consideration the hardware won’t experience all the problems the hardware might cause.
* Simulation results may be difficult to analyse, due to incomplete data.
* Some apps may run a little differently and that’s the main reason why simulators aren’t very reliable.

**Real devices Pros:**

* Test in a real environment with very real conditions, providing real insights on the functioning of the app.
* Performance of a real device is faster compared to other virtual options
* Easy to replicate bugs users are having on their devices
* Screen resolution and brightness can be easily tested in a series of different lighting scenarios
* Testing interoperability is easier

**Real devices Cons:**

* It is an expensive and time consuming solution
* There are thousands of mobile devices and having a proper testing pool that contains wide range of devices becomes really expensive
* It is difficult to get all devices that are only available in specific country.

**Conclusion:**

So both simulators and real devices has advantages and disadvantages. Based on all this factors mobile cloud testing is the best solution.