SAUCELABS

WHITE PAPER

New Perspectives on Mobile Device Testing at a Global Scale

The old days of just translating the language in your app to make it readable to people in other countries are over. Now, the scope of truly making your app relevant and usable on a global scale is much greater: in addition to the language, you have to consider the popularity of various device types in your target locations, figure out how you're going to test both legacy and newer devices, and determine a strategy that fits your resources and budget.

This whitepaper offers a realistic discussion of what's involved in testing mobile applications at a global scale today.

TABLE OF CONTENTS

4 The Newest Device is Not Necessarily the Most Popular

4 How to Accommodate Testing Legacy and Modern Devices

5 Using Real Devices vs Emulators and Simulators

6 The Benefit of Testing as a Service

7 Putting it All Together

It wasn't that long ago that the scope of localization testing for mobile devices was about ensuring language accuracy. For example, if your app was running in France, you'd want to make sure that the French that was displayed was accurate; or that apps running in South America displayed accurate Spanish or Portuguese. Apps running in Japan needed to show accurate Japanese. The language needed to match the location accurately.

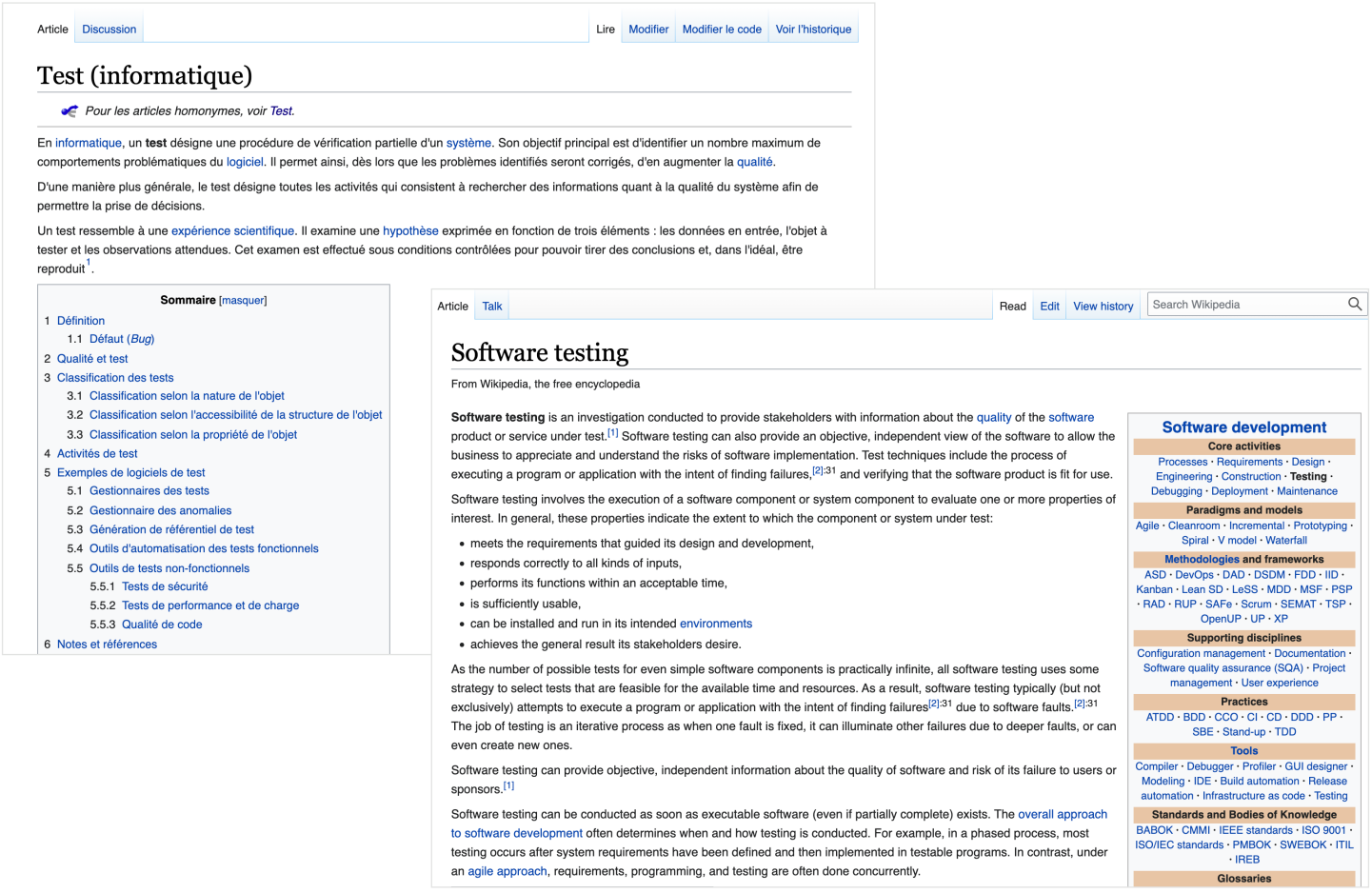

Figure 1: The global reach of the Internet has made support for multiple languages across a variety of devices commonplace

But, that was then and this is now. Today, testing mobile applications at a global scale involves a lot more than making sure that the View History tab of your web page in French does indeed display "Voir l'historique". Not only do you need to ensure that spelling and grammar are correct in the user interface, but you also need to ensure that your app works on the devices that are most common in the locales in which your app is being deployed. Your app may be displaying perfect Italian in Rome, yet keep your support department working overtime because it's failing at a lower level on older devices that proliferate the remote regions of Sicily.

The days of just making sure the language of a locale is accurate are gone and they're never coming back. A more comprehensive approach to mobile device testing at a global scale is required. A new perspective is needed.

Let's take a look at the details.

THE NEWEST DEVICE IS NOT NECESSARILY THE MOST POPULAR

Here is an interesting fact to consider: While as of this writing the current release of the iPhone is version 12, according to the website, DeviceAtlas, the iPhone 6 is the model that is used most in South Korea with an 8.45% market share.

In Brazil, older iPhones are still quite popular with models 6 to 8 taking the most overall share.

In fact, on a worldwide basis, DeviceAtlas is reporting that the iPhone 7 is the most used phone as of 2020. Also, the iPhone 7 was the most used phone in 2019 as well. Things might have changed moving into 2021, but still, it's hard to ignore the fact that while the latest model of iPhone might be perceived to be the hottest item around, on the ground it's the older models that are actually doing work.

Creating a mobile testing strategy that does not accommodate the older models is a perilous path to follow. Older phones might not be cool, but they are used in significant numbers and they need to be accommodated in your testing regime. The question is how.

HOW TO ACCOMMODATE TESTING LEGACY AND MODERN DEVICES

Testing a large variety of mobile devices, including both their current and legacy models, is important for companies that need to support a broad user base. The question is which models do you test and how far back in the model versions do you go.

Logic dictates that if a model has a significant active market share, it should be part of the testing plan. For example, does it make sense to include a model such as the Nokia 1100, which at one time was the biggest selling cell phone on the planet until it was discontinued in 2009? It does if you find that a significant number of actual and potential users of your application still use the device. Making that determination will require some compelling research. As illustrated above, you can't dismiss a cell phone model just because it's old. The decision as to what phone to use in your testing plans needs to be based on solid facts.

Then, once you determine how far back you need to go when choosing legacy devices to test, you need to figure out how to actually test the identified models. You have two choices. You can test the real devices or you can use an emulator or simulator.

USING REAL DEVICES VS EMULATORS AND SIMULATORS

Testing real devices is, as the name implies, the act of performing tests upon the physical mobile device. In this type of testing scenario, you connect the mobile device to a testing apparatus, which is usually another computer or test platform. Then you execute tests from the testing apparatus against the device. The test apparatus gathers the test results for later analysis.

Another technique is to inject the tests directly into the phone. As tests are run, they emit their result metrics back to an external device that captures test results and performs subsequent analysis.

There's a lot of value in testing against real devices directly. All the low-level peculiarities that are special to the given device are exposed. Also, if you're doing human-centric UI testing, you have the benefit of observing real-world interaction with the device. The most telling trade-off is that you need to ensure that developers and testers have access to the devices that are part of the testing array. In many cases this means that everyone has all the devices. This can get very expensive, particularly when certain legacy devices are hard to acquire. Emulators and simulators, on the other hand, provide a good deal of utility for both developers and testers at a very low cost.

An emulator is software that represents a physical mobile device. Simulators do not mimic the real device, just the software. You can think of them as virtual machines for cell phones.

Most integrated development environments (IDE) and testing tools for mobile devices come with emulators and provide the capability to add additional ones. For Mac, Xcode includes the iOS simulator. There are a number of emulators available for iOS devices that run on both OSX and Windows operating systems. Also, there are emulators for Android devices that will run on OSX, Linux, and Windows machines. In addition, there are emulators for less popular devices such as those made for Windows, Nokia, Symbian, and BlackBerry platforms.

Emulators, as well as simulators, are very useful at the developer level. Programmers can use emulator or simulator technologies to accomplish most of the higher-level work they need to do to create applications that are intended to run in a cell phone's browser or as native programs. Not only are emulators and simulators useful for phone development, but they also provide benefits for developing code for dedicated gaming devices, which have a user base of considerable size.

Emulators and simulators are also useful in testing, particularly for executing automated tests. It's all a matter of setting up your testing environment to include the emulators you need and then targeting test activities to the particular device emulators accordingly.

Still, while emulators and simulators can support most development and testing activities intended for mobile devices, they can't support all activities. You should anticipate preparing and executing a certain segment of tests against a variety of dedicated real devices. As mentioned above, real devices have peculiarities that appear in the strangest of ways; for example, natural gestures, push notifications and issues related to the SIM card, to name a few. The surest way to address such anomalies is to test directly on the real device.

THE BENEFIT OF TESTING AS A SERVICE

A comprehensive approach to testing at a global scale requires a scope of testing that focuses on application behavior in terms of both code and device. To implement this scope of testing requires a lot of work.

When using emulators or simulators, you need to make sure that your testing environment has everything required. And, as mentioned above, even if you are using emulators or simulators in most of your testing activity, you will still need to run a portion of your tests directly on real devices. This means that you're going to have to spend time and money acquiring the devices you need and then make it all available to the testing environment. This is difficult enough for current models of mobile devices but when it comes to acquiring and setting up older models, the difficulty and the expense of the work can increase dramatically.

Some companies can absorb this cost without much impact. However, for many businesses, supporting the breadth of devices required to conduct comprehensive testing at a global scale can be an excruciating expense not only in terms of the cost of the device and software involved—but also for the time it takes for your engineers to get everything up and running.

Fortunately, there is a way to alleviate the burden of cost of real devices and the expense of environment provisioning. A cost-effective way to approach mobile testing is to use a testing service otherwise known as a "cloud-based test platform."

An enterprise-grade testing service provides all the capabilities you need to conduct comprehensive testing of mobile devices at a global scale, both in terms of real devices as well as device emulators and simulators. When you use a testing service, it's just a matter of selecting the devices you want to test—either by emulation, simulation, or physical device, doing some configuration, and then running your tests against that configuration in the service's testing environment. If this sounds simple, that's because it is.

The business of an enterprise-grade testing service is to ensure that its customers have access to the devices and testing environments they need to meet their testing requirements. That's what they do and they need to do it well to stay in business. Companies that provide Testing as a Service are highly motivated to "get it right" at a competitive price point.

The Sauce Labs Continuous Testing Cloud provides a comprehensive array of iOS and Android emulators and simulators. Also, the platform supports a broad array of physical devices, including current versions of iPhone and Android devices, plus those going as far back as the iPhone 6 and the Galaxy J5, both of which were discontinued in 2018.

Using a cloud-based test platform costs money, but when you consider what you get for that service, the expense is more than justified. Just consider the number of hours it would take an in-house employee to find, set up, and make available a discontinued phone such as the iPhone 7. It's a daunting, time-consuming task. Yet as we described earlier in this article, the iPhone 7 is the most used phone in South Korea. If you want your app to work in South Korea, you need access to that phone in order to do your testing. You can either go through the time and expense required to acquire that phone on your own or you can use a service such as Sauce Labs and have access to the device immediately.

The choice seems obvious.

PUTTING IT ALL TOGETHER

State-of-the-art testing for mobile devices on a global scale requires that companies go beyond the usual activity of ensuring the quality of language translation. Given that software development has been an international activity since the first commercial mainframes appeared in the 1950s, proper language translation is expected for any piece of software that's intended for a global audience. It's low-hanging fruit.

Today more is needed. As the Internet permeates every facet of computing activity from desktop computers, to mobile devices, and even into the realm of the Internet of Things (IoT), the device on which the code is running needs to be tested in terms of the location in which it is being run. These days the latest and greatest model of a cell phone used by a college student in Silicon Valley might be a luxurious desire for a cab driver in Argentina. And yet, both might need the benefits that your software has to offer. Thus, you need to make sure that your code works on the preferred phones of each.

The best way to do this is to make sure that you have both the software and real devices available to take a global approach to comprehensive device testing. For some companies, this might mean incurring the cost that goes with testing in-house on a global scale. Yet, for many (if not most) companies, using a cloud-based test platform such as Sauce Labs is the best, most cost-effective way to implement world-class testing. Regardless of which path your company takes, the most important thing is to implement a global testing regime that makes sure your code works to the benefit of those using it no matter where on the planet they reside. After all, making code that benefits the needs of the customer is the core mission of any enterprise.

SAUCELABS

ABOUT SAUCE LABS

Sauce Labs is the leading provider of continuous testing solutions that deliver digital confidence. The Sauce Labs Continuous Testing Cloud delivers a 360-degree view of a customer's application experience, ensuring that web and mobile applications look, function, and perform exactly as they should on every browser, OS, and device, every single time. Sauce Labs is a privately held company funded by Toba Capital, Salesforce Ventures, Centerview Capital Technology, IVP, Adams Street Partners and Riverwood Capital. For more information, please visit saucelabs.com.

saucelabs.com/signup/trial

FREE TRIAL

SAUCE LABS INC. - HQ 116 New Montgomery Street, 3rd Fl San Francisco, CA 94105 USA