Mobile Application Testing - Return of the Technical Tester?

An

OZTester MAGAZINE

SUPPLEMENT

Click here to subscribe to OZTester Magazine

by Reghunath Balaraman

(Infosys Australia)

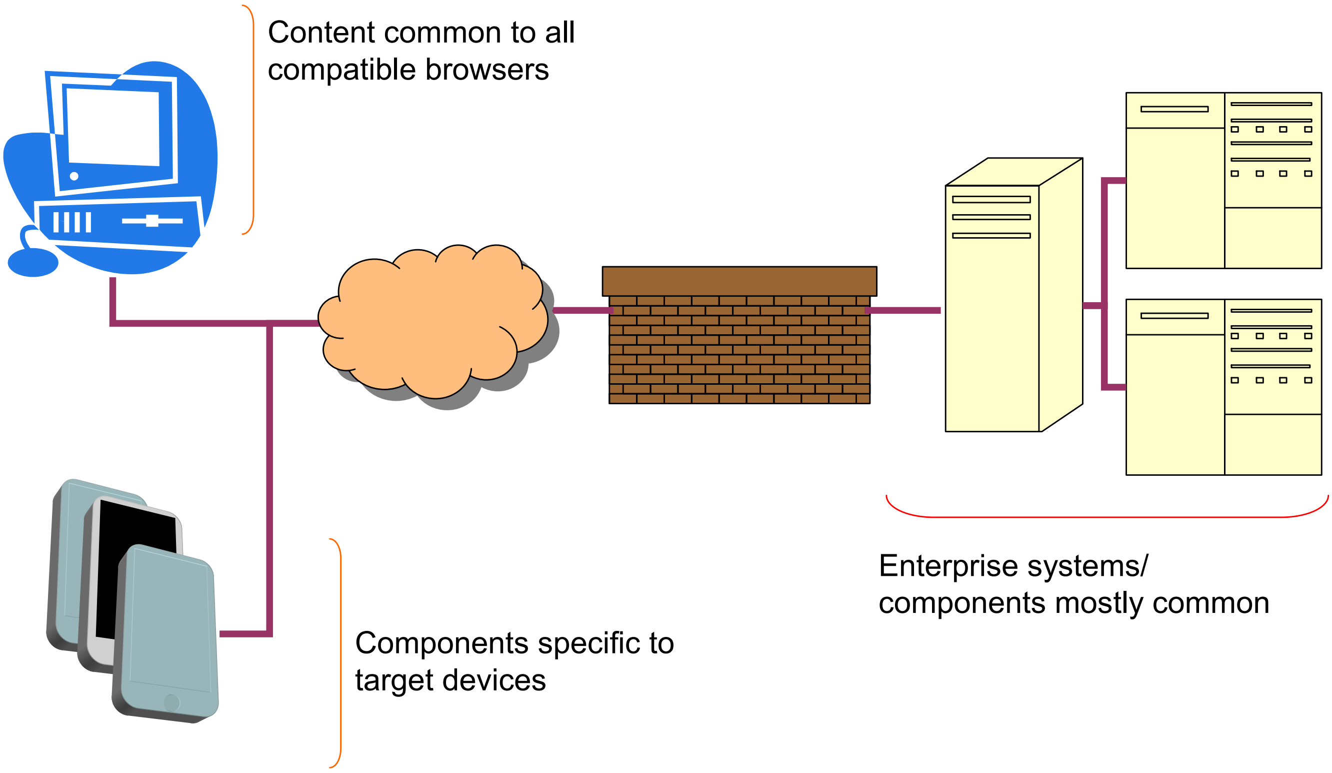
Mobile Application Testing - Return of the Technical Tester?

by **Reghunath Balaraman**

The trend in the number of mobile devices, including smart phones and tablets, is a very significant development that will have a far-reaching impact on applications development and testing. Applications are not only becoming mobile-enabled, they are also becoming an integral component in completing business transactions. Take, as an example, placing a trade in a trading system. Examining the technical skills and tests required to validate a mobile application (app) reveals that not only are there several commonalities to testing apps across domains but there is also a limited dependency on domain-specific knowledge. Are we seeing a shift in focus from the domain knowledge that made testing vertical-oriented (segmented based on industry or domain) to testing becoming a horizontal offering (spanning across industries) as it used to be? In order to answer this question, we need to deep dive into the technical aspects of testing in the world of mobile application quality assurance.

In theory, we have three broad categories of software for mobiles – the native apps, mobile web and hybrid. The native apps reside in a hand-held device whereas the mobile web accesses web content using a mobile browser. Hybrid applications utilise a combination of both hybrid and web features. While there are several commonalities between the three, they are unique in several ways.

Testing a native or hybrid application needs a different strategy and approach than testing browser-based applications on mobile devices. Functionalities ranging right from the code base to the capabilities of the native hardware will have an influence on the testing outcome. Testers need to understand and develop suitable strategies to


Content common to all compatible browsers
Enterprise systems/ components mostly common
Components specific to target devices

Mobile devices share with the web based systems, several enterprise systems and components

focus on functionalities as well as device and OS-dependant features that need to be tested on each target device.

When testing the mobile web app, the focus is primarily on the mobile browser's capability to interpret the code and deliver the content correctly and consistently. Different browsers use different engines internally. For example, Chrome uses WebKit while Microsoft uses Trident. The way they render content is not universal, especially while handling evolving standards like HTML5. Native apps reside in the device and interact more closely with the host OS and the hardware and software resources of the device. Here, testing becomes very critical as different operating systems use a different code base. Hybrid apps extend the challenges by combining both native and web characteristics into one.

Device manufacturers are stretching their creativity in an attempt to increase their market sharing by designing devices with all possible variations in form factor, processing power, user interface standards, and network capabilities. OS choices are as diverse as the hardware options. With the availability of different Android versions, device manufacturers and telecom carriers modify operating systems to optimise performance in their networks. Android currently has around 75% of the market share followed by Apple with a share of 15%. However, the market is very dynamic and the user base is not uniform across geographies. Arriving at an optimum combination of devices to test is one of the key factors for a successful test strategy for mobile devices.

The nature of the application is the main consideration while developing a test combination matrix for mobile applications. Building a target user map is the first step towards identifying the device combination. Market research can provide pointers to the end user preferences of devices and platforms. Depending on the type of the application, the dependency of the design elements on the device and OS also needs to be determined. This helps further narrow down the selection of target devices. Testing of carrier-dependent capabilities is an additional test that mobile testers need to perform. Overall, the testers need to repeat a large number of tests on a variety of devices and several operating systems and their variations.

Application development tools and mobile manufacturers provide emulators to test mobile applications. These emulators help run the code in an emulated environment to check how it will perform when it is deployed on a real device. This is a very good tool for unit testing and to some extent, the initial sanity testing. However it can never replace a real device as the emulator is a piece of software with its own limitations and cannot completely mimic the application behaviour on a real hardware. This necessitates a large investment in device inventory.

Device obsolescence and new device introduction are other factors that challenge the test combination on an ongoing basis. The current trend of major mobile manufacturers rolling out at least two new or upgraded devices every year – along with several OS updates – calls for dynamism in the test combination matrix coupled with frequent updates. In most cases, where the target devices are large, maintaining the device inventory is a large investment as well as ongoing expense. Many providers have come up with innovative solutions that help users access devices remotely, and for fixed periods, thus eliminating the need to invest and manage devices in-house.

Given the repetitive nature of the tests, automation is an obvious choice. However, it comes with its own challenges. There are several technologies that drive mobile automation from complex image recognition techniques to a combination of image and text identification. The devices and their form factors have a high level of influence on the way the user interface components are positioned and displayed. Thus, the ability to identify these components agnostic to their position and rendering is an important requirement for a successful automation tool. The learning curve needed to use the tool, its integration with test management tools, and ability to extend the existing automation in non-mobile components of the application are also of utmost importance in selecting a tool. My personal preference is to extend the existing functional automation tool rather than introducing a new tool that is aimed solely at automation of mobile components. A common tool will provide more opportunities for end-to-end automation by not limiting itself to testing the mobile functionality, and will also help extend the existing functional automation scripts to cover mobile specific tests in cases where such scripts exist.

Security is another important consideration when going mobile, especially using native and hybrid applications. These applications tend to store a variety of information at the user end, enhancing the performance and user experience. As these devices are exposed to several other applications that may have malicious intent, verifying the security of the mobile application and its interface with the enterprise systems is of prime importance. Similar to the discussion on functionality, security also needs to be evaluated in combination with the device and OS, with an extended focus on jail-broken devices which are more vulnerable to security threats.

Mobile applications have a significant part to play in enhancing the customer experience. With a good number of companies using multiple channels, including social media, providing an intuitive and consistent user experience is of paramount importance. This includes consistency in user interface design and across internet channels, ease of operations, and performance.

A common pattern that we see in the above is an independence from the functionality that is dependent on a domain or industry. While the functionality is very important, the purpose of making applications mobile-enabled; making them easier and more accessible, and providing a better user experience makes it less domain-dependant. In fact, most of the discussion above was completely domain-neutral.

It requires deep understanding of the architecture of mobile applications and design patterns, their relative merits and drawbacks, and experience in evaluating usability and performance to analyse the user experience potential of an application. This also allows extensive reusability across industries and the ability to use common tools and techniques across scenarios. Hence there is an increased focus on the technical abilities of the tester and perhaps the re-birth of the 'technical tester'. This surely will not make the domain-focused or context-driven testers obsolete. But it will emerge as a desired skill – if not establishing itself as a parallel role to the domain-focused functional tester.

Reghunath Balaraman an Enterprise QA Transformation Principal Consultant with Infosys in Sydney. He can be contacted on b.reghunath@gmail.com

Reprinted from:

OZTester MAGAZINE

Click here to subscribe