Selecting the Right Mobile Test Automation Strategy: Challenges and Principles

By choosing the tools and automation processes best-suited to testing mobile apps, QA organizations can reduce complexity and speed time to value.

Executive Summary

The mobile phone has transitioned from its traditional role as a mere communications medium to that of an essential multipurpose personal gadget. Consider that the number of smartphone users worldwide will surpass two billion in 2016, according to new forecasts from eMarketer. Meanwhile, technological advancements and the proliferation of devices across platforms and operating systems (Apple iOS, Android and Windows Mobile) have made it more challenging for hardware manufacturers and application developers to develop and roll out new products.

For vendor peace of mind, mobile applications must be tested to ensure they run on key platforms and across a multitude of networks. Despite the pressures of short mobile development cycles, it is necessary to quality-test applications across operating systems, device platforms and networks in order to ensure long-term success in a highly fragmented and competitive global market. Moreover, nonfunctional testing — including usability, security and adaptability — is as critical as functional testing. Effective testing can help device makers and application developers collect appropriate metrics that improve product quality.

This white paper explores industry best practices for mobile testing and demonstrates effective ways to manage mobile application quality through test automation. As with conventional applications, there is an increased need to automate mobile applications testing to improve scale and efficiency. A well-orchestrated automation strategy enables hardware manufacturers and software developers to reduce their efforts and accelerate time to market.

Key Challenges in Application Test Automation

The primary factor that determines an automation tool’s success is its ability to work across platforms and technology stacks. It is also important that the tool integrates lifecycle aspects such as continuous integration to enable quicker release cycles. The following challenges influence automation success:

- **Device diversity:** Multiple platforms and browsers, rendering differences, mobile devices with varied application runtimes.
- **Platform challenges:** Frequent platform updates and releases, customization of platforms by OEMs.
• App technology challenges: The diversity of app development technology; rapid oscillation of adoption between native, Web and hybrid technologies; emergence of responsive Web technologies.

• Network challenges: Multiple network types (e.g., GSM, GPRS, Wi-Fi, Wi-Max); different speeds of connectivity across geographies; multiple network operators with customized network features.

• Hardware challenges: Limitations in processing speed, limitations of mobile memory size, differences in device communication protocols (e.g., WAP, HTTP).

Test Execution Expectations
Executing tests across different devices and browsers is a cumbersome and challenging exercise. The expectations from an automation tool are limited not only in terms of cross-platform application, but also with regards to ease of usability, integration and rapid execution (see Figure 1). Some of the key imperatives include:

• End-to-end QA automation: The ability to address functional, nonfunctional and integration test scenarios.

• Scriptless automation: Test automation for mobile apps with minimal learning.

• Parallel automation: Automated test execution across multiple devices and platforms for increased ROI.

• Continuous integration (CI): Daily execution of automated scenarios in Agile and CI environments.

Mobile Testing Industry Opportunities
In the growing mobile market, the need for specialized mobile testing is growing. The following three opportunities contribute to almost 80% of the total market demand.

• Mobile automation testing: Available automation tools offer diverse capabilities for a combination of technologies. The best approach can be identified on a case-by-case basis.

• Functional testing using device labs, cloud labs: Mobile functional testing always involves a non-heuristic component that cannot be automated. However, use of device cloud labs is a widely adopted strategy that offers quicker turnaround and reduced spend on infrastructure.

• Mobile performance testing: Compared with a desktop/notebook environment, mobile device system resources (e.g., processing power, memory, etc.) are limited. Also, mobile traffic has outgrown desktop traffic in terms of online

Real-time Expectations from Smartphone-based App Automation

Figure 1
access, so performance testing of mobile applications is crucial. In addition to performance, security testing is crucial, as there is a high adoption of mobility in the banking, insurance and healthcare domains, all of which involve sensitive customer data.

Mobile Automation: What and Why?
Mobile applications are becoming increasingly sophisticated, which has significantly increased the requirement for end-to-end testing. To tackle this, test organizations are exploring alternatives to traditional manual testing. Automated testing is a highly effective approach that can offer significant business returns, as long as it is implemented using the right tools and architecture, factoring in cross-platform challenges. In such a scenario, the following specific types of testing also need to be automated (see Figure 2):

- **Cross-platform compatibility testing**: This is necessitated by the growing number of handsets and platforms.
- **User experience testing**: Most testing organizations have limited experience with the design and execution of usability tests.
- **Field or network testing**: Testing must be performed in a geographically distributed environment to account for the variety of network types.
- **Location-based testing**: These are tool sets that automate application response across various locations.
- **App store certification**: This is a pre-certification of repeated app store checks required for leading app stores.

Test Automation Mechanisms for Mobile Apps
A mobile app or responsive Web site rendered on mobile can be automated in one of the following ways:

- **Web-based automation**: This approach is specific to the mobile Web. It leverages user agent add-ons that come with popular browsers such as Mozilla Firefox and Google Chrome. These user agents help render the specific Web content that is displayed on the device onto a regular desktop browser. The functionality can be automated by using license-based and open source solutions.
- **Cloud-based automation**: Cloud-based test automation vendors support Web-based QA across platforms. Cloud-based solutions also offer integration with open source automation libraries.
- **Lab-based automation (devices and simulators)**: This method involves setting up tools in a test environment and leveraging simulators or physical devices to automate testing using popular commercial and open source tools.

Additional Areas of Automation in End-to-End Mobile Application Testing

Figure 2
Open Source and Platform-specific Tools
The reliability of device platform-specific tools is a high priority. While platform vendors and open source tools prove to be good options, clients require additional programming skills to leverage them effectively.

Cognizant Tools
We offer a built-in platform to easily automate testing of mobile native and Web apps, and perform end-to-end automation across platforms and devices, simulators and cloud topologies. The platform enables scriptless automation, as well as seamless integration with device cloud and CI systems, thereby ensuring world-class mobile QA.

Mobile App Testing Tools Selection Criteria
There are many ways to improve the effectiveness and efficiency of testing applications on mobile devices. It is essential to ensure functional and nonfunctional compliance of mobile applications on devices and mobile automation. It is also important to establish labs for testing mobile applications. Some basic criteria for mobile app testing include the use of emulators, multiple browsers, different platform versions and mobile-specific scenarios for GUI, field and interruption testing.

One best practice is to start with a proof of concept (PoC) on a critical platform prior to narrowing down to a particular tool or toolset, as one tool may not achieve your cross-platform automation goals. When test automation tools are evaluated, organizations must not overlook the integration of test management with device cloud environments.

A starting point is to identify reusable scenarios across platforms, group them accordingly and map each appropriate phase in the app development lifecycle to leverage the benefits of the appropriate and time-saving framework. For mobile apps, critical test cases can be run across releases as automation candidates to ensure cross-platform reusability and multi-browser compatibility. It’s important to consider the different form factors when determining test coverage, especially for the mobile Web. Also, consider the different aspects of connectivity for automation execution runs.

In addition, there are 10 key principles that need to be considered when selecting the right tool:

1. Perform tool feasibility tests, as mobile technologies and platforms vary.
2. Select tools that support both platform simulators and devices, because you can mix and match devices and simulators to optimize runs on different platforms.
3. Aim for automation in nonfunctional areas, such as interruption and hardware scenarios that include battery state changes.
4. Always optimize platform support; in some cases, there may be a need for one or more tools to perform automation.
5. Identify multiple device and version support.
6. Ensure ROI for each of the mobile platforms and duration of execution as a part of the tool consideration process.
7. Look for avenues for early automation because most mobile projects are Agile.
8. Ensure integrated execution with the test management tool. Application lifecycle management is an important success factor.
9. Look for data-driven automation support; iterations in execution will increase coverage and ROI.
10. Good candidates for automation are test cases that have repeated business logic, functional flows or data-driven tests, such as smoke testing and build acceptance testing, regression testing, synchronization testing, compatibility testing, GUI testing and non-dynamic UI components, as well as cross-platform functional testing.

Moving Forward
The principles and success factors of test automation for mobile applications are much different from those applied traditionally to conventional applications. The attributes of devices, environment, toolsets and test coverage add layers of complexity. This requires keen insight and a customized approach for a well-orchestrated automation strategy.

Using these best practices, organizations can set test automation goals and choose the appropriate coverage and toolsets for a comprehensive approach. Making automation effective and efficient yields cost and time reductions in the app development lifecycle.
About the Author

Ramakrishnan Venkatasubramanian (Ram) is a Technology Leader in Cognizant’s Quality Engineering & Assurance Practice. As a part of the Technology Center of Excellence, Ram leads Digital Technology, including R&D, mobility QA, service virtualization and the Internet of Things QA. Ram has over 15 years of experience in the information technology industry, with experience spanning software development, product engineering, test automation, mobility, cloud, the Internet of Things and technology consulting. Ram can be reached at Ramakrishnan.Venkatasubramanian@cognizant.com.

The author would like to thank Pradeep Kumar Govindasamy for his contributions to this white paper.

About Cognizant’s Mobile Testing CoE

Cognizant's Mobile Testing Center of Excellence (CoE) has played a pivotal role in delivering strategic mobile testing programs for 350+ customers. With a wide range of mobile technologies, the practice specializes in mobile consulting, research on cutting-edge mobile technology solutions, developing frameworks, solution accelerators and best practices for end users. The Mobile Testing CoE features a strong pool of certified mobile testing experts, built using long-term competency development programs. In addition, it houses new tools and technology powered by a focused R&D team, which takes mobile application testing to a whole new dimension.

Footnotes