

Testing Mobile Business Applications

A hybrid approach to testing mobile business applications that combines manual testing with automation testing



EXECUTIVE SUMMARY

The phenomenal growth of mobile devices has opened up avenues for organizations to integrate them into the mainstream computing environment. Today's mobile applications deliver complex functionality on platforms that have limited resources for computing. Yet, unlike the PC-based environment, the mobile environment comprises a plethora of devices with diverse hardware and software configurations and communication intricacies. This diversity in mobile computing environments presents unique challenges in application development, quality assurance, and deployment, requiring unique testing strategies.

Many enterprise applications that were deployed as desktop applications are now being ported to mobile devices. These applications are empowering workforces across various functions, especially those in sales, supply chain, field support, and on the shop floor.

Mobile applications, although they have limited computing resources, are often built to be as agile and reliable as PC-based applications. In order to meet the challenge, mobile application testing has evolved as a separate stream of testing.

In this document, we outline a hybrid approach to testing mobile business applications that combines manual testing with automation testing using emulators and devices.

"The market size for mobile business applications will grow to about \$5.8 billion by 2008."

Source: "Applications for Mobile Information Devices" by Sun Microsystems



TABLE OF CONTENTS

Unique Challenges in Testing	4
Mobile Application Testing	4
Weighted Device Platform Matrix	5
Guidelines for Testing Mobile Applications	6
Conclusion	6
ABOUT KEANE	7

TABLE OF FIGURES

Figure 1 Keane’s Recommended Strategy for Testing	5
Figure 2 Weighted Device Platform Matrix	6



Unique Challenges in Testing

Mobile business applications can be classified into standalone applications and enterprise applications. Standalone applications reside in the device and do not interface with external systems. Conversely, enterprise applications are built to perform resource-intensive transactions that are typical of corporate computing environments. Enterprise applications also interface with external systems through Wireless Application Protocol (WAP) or Hyper Text Transfer Protocol (HTTP).

The unique challenges in testing mobile applications arise from the diversity of the device environment, hardware and networking considerations, and Rapid Application Development (RAD) methodologies. These challenges are outlined below:

Diversity of the Device Environment

The realm of mobile computing is composed of various types of mobile devices and underlying software (hundreds of device types, more than 40 mobile browsers). Some of the unique challenges involved in mobile testing as a result of this condition are:

- Rendering of images and positioning of elements on the screen may be unsuitable in some devices due to the difference in display sizes across mobile devices and models. Exhaustive testing of user interfaces is necessary to ensure compatibility of the application.
- Mobile devices have different application runtimes. Binary Runtime Environment for Wireless (BREW), Java, and embedded visual basic runtime are just some of the runtimes commonly available in mobile devices. Applications should be tested exhaustively for the variations specific to runtime.

Hardware Configuration & Network-related Challenges

The mobile environment offers lesser memory and processing power for computing when compared with the traditional PC environment. Unlike

the network landscape of the PC environment, the network landscape of a mobile device may have gateways (access points between the wireless internet and the cable internet). Some of the drawbacks of diverse hardware configurations and the network landscape of mobile devices are:

- Limitations in processing speed and memory size of mobile devices lead to variations in performance of applications across different types of devices. Testing programs should ensure that the applications deliver optimum performance for all desired configurations of hardware.
- Some devices communicate through WAP while some others use HTTP to communicate. Applications should be tested for their compatibility with WAP-enabled as well as HTTP-enabled devices.
- Network latency (time taken for data transfer) will be unpredictable when applications communicate over network boundaries, leading to inconsistent data transfer speeds. Testing should measure the performance of applications for various network bandwidths.
- Gateways in a wireless network may act as data optimizers that deliver content more suitable for specific devices. This data optimization process may result in decreased performance for heavy traffic. Testing should determine the network traffic level at which gateway capabilities will impact the performance of the mobile application.

Rapid Application Development (RAD) Methodologies

In order to deliver the benefits of faster time to market, RAD environments are used for mobile application development. Since the time taken for development is reduced by the introduction of RAD tools, builds will be available for testing much earlier. Therefore, a RAD methodology imposes an indirect pressure on testing teams to

Keane Facts

Keane has experience in testing mobile applications that are distributed globally.

Keane has experience in performing mobile application testing on as many as five devices.

Keane has dedicated T1 lines to connect offshore development centers to our customers' environment.

reduce the testing cycle time without compromising quality and coverage.

Mobile Application Testing

The critical factors that determine the success of a mobile testing program are:

- Use of test automation
- Use of emulators and actual devices
- Testing for mobile environment and application complexity

In this section, we discuss these critical success factors and present our recommendations for mobile application testing.

Use of test automation

Testing of mobile applications is traditionally done by manual execution of test cases and visual verification of the results. However, it is an effort-intensive and time-consuming process. Automating the appropriate areas of a testing program can yield quantifiable benefits.

Use of emulators and actual devices

Emulators can be beneficial for testing features of the application that are device independent. However, actual devices should be used for validating the results.

Testing for mobile environment and application complexity

Due to diversity in mobile hardware and platforms, testing programs need to incorporate GUI and compatibility tests



Approach	Type of Testing	Manual Testing		Automated Testing on the Devices
		Using Devices	Using Emulators	
Standard Testing	Unit Testing	NO	YES	NO
	Integration Testing	NO	YES	NO
	System Testing	YES	NO	NO
	Regression Testing	YES	NO	YES
Special Types of Testing to Address Specific Challenges	Compatibility Testing	YES	NO	YES
	GUI Testing	YES	NO	NO
Types of Testing More Relevant for Enterprise Mobile Business Applications	Performance Testing	YES	NO	YES
	Security Testing	YES	NO	YES
	Synchronization Testing	YES	NO	No

Figure 1 Keane’s Recommended Strategy for Testing

Source: Keane, 2007

in addition to the standard functionality tests.

Keane Testing Tip

Involve end users as early as possible in the testing cycle – early feedback helps in design correction.

Enterprise applications are more complex in both functionality and architecture. Such applications require performance testing, security testing, and synchronization testing in addition to the standard functionality testing.

Weighted Device Platform Matrix

End-to-end testing of an application may trigger thousands of defects for a few million lines of code. Detecting

these defects will require innumerable test combinations and input values. Repeating the test cases over many hardware and software combinations will increase the tedium of test execution. In order to optimize the effort spent on testing for various combinations, a Weighted Device Platform Matrix method may be adopted. The matrix is prepared in two steps:

1. Defining the parameters of importance

The factors that influence the importance of a specific hardware and software combination must be identified. Usually, this information will be obtained from the business requirements. Some of the factors that influence the importance of this combination are:

- Total number of users for a device and Operating Systems (OS)

- Recommendation of business to conduct test for a particular device or OS

These factors are weighed and then relative weights are assigned to each of the devices and OS.

2. Preparing the matrix for all possible combinations

A matrix, which represents the score for each combination, is prepared. The score is calculated as the product of relative weights of devices and OS. A higher score indicates higher criticality of the combination.

Based on the criticality of the combination, the required degree of coverage can be determined.



Device \ OS	OS	Relative Importance	Windows CE	Windows Mobile Ver. 2003	Windows Mobile Ver. 2005	Palm OS	Symbian OS
Relative Importance			5	6	9	6	4
Symbol PPT 8800	7	35	42	63	NA	NA	
HP iPAQ 2100 series	9	45	54	81	NA	NA	
Tungsten E2	5	NA	NA	NA	30	20	
Tungsten T5	6	NA	NA	NA	36	24	
HP iPAQ 1900 series	7	35	42	63	NA	NA	

High Importance

Low Importance

Medium Importance

Combination Not Available

Figure 2 Weighted Device Platform Matrix

Source: Keane, 2007

Conclusion

Organizations intending to deploy mobile applications must plan their testing strategy across manual and automation testing approaches for efficient and error-free delivery. In addition to actual device-based testing, emulators should be included as an integral part of the testing program. Enterprise applications require special testing techniques. Outsourcing to vendors who are operating an independent testing practice may be a viable option to manage the expertise, scalability, and quality assurance requirements for mobile application delivery.

Guidelines for Testing Mobile Applications

- 1 Understand the network landscape and device landscape before testing to identify 1 bottlenecks.
- 2 Conducting testing in uncontrolled real-world test conditions (field-based testing) is 1 necessary, especially for a multi-tier mobile application.
- 3 Select the right automation test tool for the success of the testing program. Rules of thumb for an ideal tool are:
 - One tool should support all desired platforms.
 - The tool should support testing for various screen types, resolutions, and input mechanisms — such as touchpad and keypad.
 - The tool should be connected to the external system to carry out end-to-end testing.
- 4 Use the Weighted Device Platform Matrix method to identify the most critical hardware/ platform combination to test. This method will be very useful especially when hardware/ platform combinations are high and time to test is low.
- 5 Check the end-to-end functional flow in all possible platforms at least once.
- 6 Conduct performance testing, GUI testing, and compatibility testing using actual devices. Even though these tests can be done using emulators, testing with actual devices is recommended.
- 7 Measure performance only in realistic conditions of wireless traffic and user load.



About Keane

Keane partners with businesses and government agencies to *optimize* IT investments by delivering exceptional operation, maintenance, and evolution of mission-critical systems and business processes. Keane helps clients realize the greatest value from their IT investments by leveraging an insider's hands-on understanding of the nuances and subtleties of their applications, processes and infrastructure making the recommendations we give more actionable, the work we do more pragmatic, and the results realized more measurable.

In business since 1965, Keane is an agile, mid-sized, full service IT services firm with headquarters in the United States and more than 13,000 employees globally.

For more information on Keane's services, solutions, and locations, please visit www.keane.com.



Keane

Corporate Headquarters
88 Kearny Street, Suite 1650
San Francisco, CA 94108

For more information about Keane's services, contact us at:

877.88.KEANE
info@keane.com
keane.com

