

# Criteria for Selecting Mobile Application Testing Tools

BOŠTJAN ARZENŠEK and MARJAN HERIČKO, University of Maribor

---

The importance of software testing has been discussed and proven in many articles and in existing research. Software testing plays a key role in providing quality software solutions and services. Standard testing approaches and methodologies were adequate until the arrival of mobile technologies. With mobile technologies, the testing process was forced to change in the face of significant challenges, the most important one being mobility. Mobility provides a pallet of challenges that are unique and demand new testing approaches and methodologies in software testing. The identification of challenges and issues has helped the development of the mobile software testing process and tools. With a wide range of new testing tools, testers face a new challenge in selecting the right tool and methodology for testing mobile applications. In this paper, we will present criteria for selecting mobile application testing tools based on identified challenges and issues, testing approaches and strategies. We will provide a proposal for a simpler and quicker way of selecting the appropriate tool for testing mobile applications.

General Terms: Mobile applications testing

Additional Key Words and Phrases: testing, mobile applications, mobile technologies

---

## 1. INTRODUCTION

The increased use and the rapid development of mobile devices and technology is a clear sign of future trends in ICT. Not only do the number of mobile users increase daily, the same thing is occurring in the market for tablet PCs. According to Gartner, the number of purchased tablets is going to surpass the number of desktop-based computers in the beginning of the year 2015 [Rivera and Van der Meulen, 2013]. More and more people are using mobile technologies in their everyday lives for interaction, entertainment, business and more. With mobile applications we can extend the usability of mobile devices even further. A mobile application is an application that runs on a mobile device and is context aware. This means that the application is aware of the computing environment in which it runs and can adapt/react according to its current context [Muccini, 2012]. Context-awareness is just one of the many challenges in mobile application testing, which demands new methods and testing approaches. There are many challenges in mobile software testing, which by definition [Gao et al. 2013] means: “testing activities for mobile-based applications and mobile web applications on mobile devices using well-defined software test methods and tools to ensure the quality in mobile service functions, behaviors, performance and QoS, as well as mobile features, such as mobility, usability, inter-operability, mobile connectivity, security and privacy.” Mobile applications that are free of faults and errors provide a better user experience, which has a direct impact on the business success of the application. Users grade the quality of the mobile application based on their user experience. Unfortunately, many new users choose applications based on previous reviews and grades. Therefore, old errors and faults, or a poor user experience in an otherwise working application can lead to the business failure of the application.

In this paper we will present the criteria for selecting mobile application tools based on the identified challenges in mobile testing, testing approaches and strategies. First, in Section 2, we will present the main challenges and issues in mobile software testing. In Section 3, we will present the four testing approaches that have been identified by [Gao et al. 2013]. In Section 4, we will present and analyze related work on mobile testing tools, which provides the basis for our criteria definition. Our main focus is to present the criteria definition process and to extend the criteria for selecting mobile testing tools that have been identified by [Gao et al. 2013]. The idea is to have a selection of criteria that can be used for the characterization of a variety of mobile testing tools. Once we have the mobile testing tools characterized

---

Author's address: B. Arzenšek, Faculty of Electrical Engineering and Computer Science, University of Maribor, Smetanova 17, 2000 Maribor, Slovenia; email: bostjan.arzensek@um.si; M. Heričko, Faculty of Electrical Engineering and Computer Science, University of Maribor, Smetanova 17, 2000 Maribor, Slovenia; email: marjan.hericko@um.si.

*Copyright © by the paper's authors. Copying permitted only for private and academic purposes.*

In: Z. Budimac, T. Galinac Grbac (eds.): Proceedings of the 3rd Workshop on Software Quality, Analysis, Monitoring, Improvement, and Applications (SQAMIA), Lovran, Croatia, 19.-22.9.2014, published at <http://ceur-ws.org>

properly we can make a quicker and more effective selection. In the Discussion, we will comment on our findings.

## 2. CHALLENGES OF MOBILE TESTING

There are many challenges in mobile software testing due to the nature of the environment in which the mobile applications are running. Based on the definition by [Muccini 2012]: “A mobile application is an application that runs on mobile device with limited resources. It uses the data from the surrounding environment in which the mobile device is in and/or from user actions, producing context-based output.” Mobile applications differ from traditional desktop application in many ways. In this section we will describe the main challenges in mobile software testing and why the traditional tools and methodology of software testing are not adequate.

### 2.1 Mobile Connectivity

One of the more important challenges in mobile software testing is the connectivity of mobile devices with various mobile networks and devices. Unlike desktop applications, which use fixed network connections, mobile applications connect to mobile networks, which can vary in speed, security and reliability [Kirubakaran and Karthikeyani 2013]. Usually the types of mobile networks are 2G, 3G, 4G and various wireless networks. Mobile applications rely heavily on mobile networks, which is why the challenge of mobility can have an impact on: reliability, performance, security and the correct operation of the application and/or its functionalities [Muccini 2012]. The nature of the challenge demands testing in different environments. The mobile applications are tested in:

- environment with a constant connection to the mobile network,
- environment with a variable connection to the mobile network and
- environment without a connection.

Based on the difficulties and requirements of the testing procedures, different testing approaches are recommended [Tharakan and Jacob 2012]. Because the application’s reliability, performance, security and correct functioning strongly depend on the available connection type, functional and extra functional testing has to be performed in different connectivity scenarios and networks [Muccini 2012].

### 2.2 Resource constraints

Mobile applications use the resources of mobile devices, which are very limited. Despite the rapid development of mobile devices, it is important that the consumption of resources is monitored and controlled at all times [Muccini 2012]. The resources of mobile devices include: the central processing unit, RAM, memory, touch screen, battery, as well as different modules and sensors. During the testing process, we focused on the central processing unit, RAM and memory. Because the battery and the screen constitute a different set of challenges, we treated them individually. The central processing unit, RAM and memory are components of the SoC (System-on-a-Chip) which includes other controllers and components that form a complete system [Yeap, 2013].

The excessive use of resources can reduce the performance of mobile devices and can cause malfunctions in the mobile application. During the testing process the consumption of resources must be constantly monitored.

### 2.3 Autonomy

Mobile devices need energy to run. The use of mobile devices depends on battery capacity and the way the device is used. All the device’s resources and activities use energy but not equally. GPS sensors, data transfer and video editing are activities that use more energy than others [Tharakan and Jacob 2012]. These activities use multiple device resources or require a continuous data connection, which is the main reason for higher energy consumption. Different activities have a different impact on autonomy and during the testing process all have to be monitored [Muccini 2012].

## 2.4 Diversity of user interfaces

Mobile operating systems have different user interfaces, which are defined by rules and guidelines. The use and layout of elements is checked in the verification process when publishing the mobile applications on the markets. Non-compliance with rules and guidelines can delay the publishing process, increase the cost of development and testing. Different screen sizes can also have an impact on the look and usability of the mobile application. Different mobile devices can react differently to the same application code, which must be tested with GUI testing [Muccini 2012].

## 2.5 Context awareness

Context is by definition [Abowd et al. 1999] any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the users and applications themselves.

Context can limit or extend the operation of mobile applications or its functionalities with data from the environment in which it is in. Mobile applications can be in different contexts with different data. This creates a unique challenges in the testing process [Schulte and Majchrzak, 2012].

Context aware mobile applications adapt and evolve based on the data obtained from the environment. This evolution can happen in real time without interrupting or stopping the operation of mobile applications. Unfortunately, this can lead to unexpected and unplanned changes in the mobile application's operations. The reliability of a mobile application depends on the management of context adaption. To insure the correctness of applications operation, context-specific test selection techniques and coverage criteria have to be produced [Muccini 2012].

## 2.6 Diversity of mobile devices

There are many different mobile devices, made by different vendors, which have different hardware and software settings. The number of variations is even larger if we add all the devices that have a modified mobile operating system. The vendors modify the operating system to create a better user experience for the user, or increase the functionalities of a device. Due to these variations, mobile applications can run and behave differently [Muccini 2012]. The diversity of mobile devices can also increase the costs and duration of the testing process. If we would want to test across all devices, the buying and maintaining costs of mobile devices would be enormous. If we take into account the time spent for testing, the complexity of the challenge increases. Testing techniques that maximize the diversity coverage while minimizing the devices being tested need to be devised [Muccini 2012].

## 2.7 User experience

The user experience includes the user's perceptions and feelings before, during and after the interaction with the mobile application. Often, users assess the application based on their user experience, therefore the appropriate user experience is critical for the success of the application. The adequacy of the user experience cannot be directly tested because of the subjective nature of the entire process. But we could check the correctness of individual segments and determine compliance with good practices [Knab 2012].

The adequacy of the user experience includes verification of elements and activities in the areas of design graphical user interfaces, interaction and usability of the application itself.

The design of the graphical interface is evaluated based on the proper and logical use of layouts, navigation between screens, layout of graphical elements, fonts and text [Knab 2012].

## 2.8 Touch screens

Mobile devices use touch screens as the primary means of interacting with the user. Touch screens enable the display and input of data as individual values or as a group of data. The user activates the interaction with a touch of the screen, which can be a single touch or a multi touch interaction. There are also gestures that include different sequences and combinations of touches. Gestures allow additional functionalities in the mobile application's operation, which creates new challenges in the testing process. Touch screens are tested based on the correctness of the displayed data and the responsiveness

[Kirubakaran and Karthikeyani 2013]. The responsiveness of the touch screen represents the elapsed time between the touch of the screen and the moment, when the touch is recognized, which triggers an update on the screen [Agawi App Glimpse n.d.]. The responsiveness of the touch screen is also dependent on the mobile device's resources.

## 2.9 New programming languages and mobile operating systems

Programming languages for mobile applications have been designed to support mobility, resource management and new graphical user interfaces. Traditional testing techniques do not take into account the operation of programming languages in mobile operating systems, so they need to be adjusted accordingly. To analyze the code it is necessary to be aware of the specifics of the programming languages and how they operate [Kirubakaran and Karthikeyani 2013].

Mobile operating systems are new and still only partially reliable. Various reliability issues and defaults are corrected in new versions of the operating system which are frequent but not always backward compatible [Muccini 2012]. For example, mobile devices with Microsoft mobile operating system Windows Phone 7 were not updatable with the new release of Windows Phone 8 [Hamblen 2009].

## 3. MOBILE TESTING APPROACHES

Based on [Gao et al. 2013] there are four testing approaches in mobile testing. These approaches are:

- emulation-based testing,
- device-based testing,
- cloud testing and
- crowd-based testing.

Each of them is designed to handle challenges in mobile testing, identified in the previous section, but none of them can handle them all. It is important to select the correct approach based on the functionalities of the mobile application and the challenges that they provide. Each approach has its features and limitations, which have to be identified before the selection is made.

### 3.1 Emulator-based testing

The emulation-based testing approach involves using a mobile device emulator, which creates a virtual machine version of a mobile device for inspection on a personal computer. It is often included with a mobile platform's software development kit. It is relatively inexpensive because no testing laboratory is needed and no physical devices have to be purchased or rented, but it can only be used to assess system functionality within a very limited context. Although this approach is low-cost, it has several limitations – for example, it has difficulty validating a full set of gestures because most emulators support very limited gestures and device-specific functions. Another challenge is its limited scale for testing QoS. To overcome these problems, a simulation-based approach can create a mobile test simulator to mimic various mobile client operations and support more than one mobile client. However, even this workaround has its limitations in validating device-specific mobile service functions. In addition, it is impossible to deal with diverse devices and mobile platforms because emulators are usually based on a specific device platform [Gao et al. 2014].

### 3.2 Device-based testing

The device-based testing approach requires setting up a testing laboratory and purchasing real mobile devices, which is more costly than emulation-based approaches but can verify device based functions, behaviors, and QoS parameters that other approaches cannot. In addition, it also has the advantage of being able to validate its underlying mobile networks via reconfiguration and selections in a testing environment. One of the major challenges with this approach is the problem it has in coping with rapid changes in mobile devices and platforms. Another challenge is its limitations related to system QoS because large-scale testing require many mobile devices, which is usually impossible for enterprises [Gao et al. 2014].

### 3.3 Cloud testing

This approach, based on testing through the cloud, is typically supported by testing vendors. The basic idea is to build a mobile device cloud that can support testing services on a large scale. This approach addresses the significant increase in demand for mobile testing services by using the pay-as-you-go business model. It also allows different mobile users to provision their required testing environment via a rental service model. Compared with other approaches, this can be more cost-effective than device-based testing for large-scale applications, and is much more effective for supporting diverse testing activities on mobile devices.

### 3.4 Crowd-based testing

The crowd-based testing approach involves using freelance or contracted testing engineers or a community of end users such as uTest ([www.utest.com](http://www.utest.com)), along with a crowd-based testing infrastructure and service management server to support diverse users. Currently, a service vendor supports primitive test management, a testing service, and bug reporting. Most mobile test operations are managed in an *ad hoc* way with very limited mobile test automation tools. This approach offers the benefit of in-the-wild testing without the need to invest in a laboratory or purchase or rent devices, but at the risk of low testing quality and an uncertain validation schedule.

## 4. RELATED WORK

Recent studies on mobile testing tools primarily focus on GUI-based testing, test automation and white-box testing. There are different studies that present solutions for different testing approaches and strategies, which has created numerous tools for mobile testing [Gao et al. 2014]. If we want to make an efficient selection, we must first characterise these tools and then evaluate them. The process of evaluation has many challenges. One of the reasons for this is the lack of standards, test models and coverage criteria that address the distinct requirements of mobile application testing [Gao et al. 2014]. One of the challenges is also defining the criteria with which these tools are evaluated. In [Gao et al. 2014] the testing tools are compared based on different criteria, which are listed in Table I. This comparison is a good example of a possible mobile testing tool characterization.

Table I. Criteria defined in a mobile testing tools comparison [Gao et al. 2014]

Criteria	Values
Testing strategy	GUI-based function testing
	Performance testing
	Load testing
Mobile testing tool platform	Linux
	Windows
	Mac
Mobile application platform	Android OS
	iOS
	Windows OS
Mobile application type	Native apps
	Web apps
Testing approaches	Emulation-based testing
	Device-based testing
Test properties	Supported script language
	Record and Play
License	Open source
	Subscription

The criteria are defined based on the current functionalities of the testing tools and the identified testing strategies and approaches.

Although the provided criteria allow a basic characterization of the testing tools, we believe that the criteria should be extended and created based on challenges in mobile testing. With a more extended definition of criteria, the selection process could be faster and more efficient.

## 5. CRITERIA FOR SELECTING MOBILE APPLICATION TESTING TOOLS

Based on related work described in the previous section and the identified challenges, we propose a definition of criteria that is based on these challenges and the testing strategies. Our goal is to create more detailed criteria for selecting the mobile testing tool.

The definition process consists of three phases and is illustrated in Figure 1.

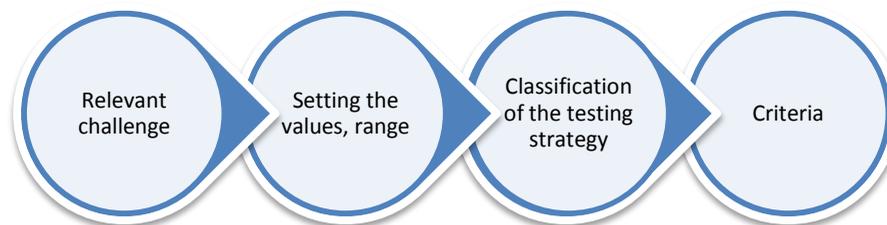


Fig. 1. The definition process of a criteria

The process begins with the selection of the mobile testing challenge, which we have identified in Section 2. The relevant challenge is analyzed from the perspective of the mobile application and based on this result, the values or the range are set. Values and range can differ depending on the type of the challenge. For example, the values for screen dimensions are small, normal, large and extra large, where all the sizes are based on [Google 2014], and values for the challenge of Bluetooth connectivity, where the values are only enabled or not enabled. After the values have been set, it is important to classify the upcoming criteria into a correct strategy. The testing strategy defines the testing activities and the goals of the testing process. Finally, after the testing strategy is defined, the criteria is created. The defined criteria is used to characterize the testing tools. Before the evaluation of the testing tools can be done, the selection of criteria must be set. The selection criteria are set based on the mobile application functionalities and on the selected testing approach. When the testing tools are selected and the criteria are set, the evaluation process can be begin.

Based on the process of creating criteria we propose a list of criteria that are defined based on the challenges described in section 2. The proposed criteria are shown in Table II.

Table II. Proposed criteria that are defined based on the challenges

Challenge	Properties	Values, range	Testing strategy	Supported feature
Mobile connectivity	Mobile network	Constant, partial, none	Functional testing	Supports changing the consistency of the mobile network
	Data transfer speed	Range of speeds (2G, 3G and 4G)	Connectivity testing	Supports changing or limiting the data transfer speed
	Bluetooth	Enabled, not enabled	Functional testing	Supports Bluetooth connectivity
	NFC	Enabled, not enabled	Functional testing	Supports NFC connectivity
	Wi-Fi	Enabled, not enabled	Functional testing	Supports Wi-Fi connectivity

	Wi-Fi Direct	Enabled, not enabled	Functional testing	Supports Wi-Fi Direct connectivity
Resource constraints	CPU	1 core, 2 core, 4 cores	Performance testing	Supports changing or limiting the operation of CPU cores
	CPU Speed	Range of speeds (1Mhz – 2500Mhz)	Performance testing	Supports changing or limiting the operation of CPU speed
	RAM	Range of values (16Mb,32Mb,64Mb, 128Mb,256Mb,512Mb, 768Mb,1Gb,2Gb,3Gb and 4Gb)	Performance testing	Supports changing or limiting the amount of RAM
	Memory	Range of values (16Mb,32Mb,64Mb, 128Mb,256Mb,512Mb, 768Mb,1Gb,2Gb,3Gb,4Gb,8 Gb,16Gb,32Gb, 64Gb,128Gb)	Performance testing	Supports changing or limiting the amount of memory
Autonomy	Consumption	Percentage of the total battery capacity	Load testing	Supports monitoring the battery consumption
	Duration	Time of the total battery capacity	Load testing	Supports monitoring the battery duration
Diversity of user interface	Guideline checker	Mobile platform specific Rules and guidelines	Usability testing	GUI guideline checker
Context awareness	GPS	Simulated, real data, not enabled	Functional testing	Simulate data from the GPS
	Neighbor devices	Simulated, real data, not enabled	Functional testing	Simulate data from the neighbor device
	Altitude	Simulated, real data, not enabled	Functional testing	Simulate data from the barometer
	Brightness	Simulated, real data, not enabled	Functional testing	Simulate data from the light sensor
	Temperature	Simulated, real data, not enabled	Functional testing	Simulate data from the temperature sensor
	Context awareness	Simulated, real data, not enabled	Functional testing	Simulate data from the environment and the user
	Context adaption	Enabled, not enabled	Functional testing	Enables data input from the context in real time
Diversity of mobile devices	Vendor and model	Enabled, not enabled	Functional testing	Simulation of a specific mobile device
	Operating system	Android, iOS, BlackBerry, Windows Phone 7 and 8	Functional testing	Supports changing mobile device platform
	Operating system versions	Enabled, not enabled	Functional testing	Supports changing mobile device platform to different versions
	Screen dimensions	Small (at least 426dp x 320dp), normal (at least 470dp x 320dp), large screen (at least 640dp x 480dp), extra large screen (at least 960dp x 720dp)	Usability testing	Supports changing screen size
User experience	Layout checker	Enabled, not enabled	Usability testing	Checks the use of layouts for specific mobile operating systems
	Text visibility	Percentage of the characters displayed based on the total number of characters	Usability testing	Checks text visibility
	Text grammar	Supported, not supported	Usability testing	Supports internationalization
	Notifications	Enabled, not enabled	Usability testing	Supports notification management
	Interruptions	Enabled, not enabled	Usability testing	Supports interruptions

				management
Touch screens	Responsiveness	Enabled to measure, not enabled to measure	Usability testing	Supports measuring the responsiveness of the screen
	Gestured	Enabled, not enabled	Usability testing	Supports gesture recognition
	Multi touch	Enabled, not enabled	Usability testing	Supports multi touch recognition

## 5. DISCUSSION

The proposed criteria and its values were set based on the defined challenges in mobile testing. The challenges in mobile software testing vary from traditional ones because of mobility. Mobility has changed the operation of applications, devices and our interaction with these devices. Consequently, mobility has changed the software testing process, which is nevertheless still evolving. Because of this, there is a possibility that the proposed criteria can change, adapt or possibly be removed from the list. Future research will confirm or refute this claim. Also, in the future we expect the development of new tools and techniques that will enable more effective mobile testing. New tools will enable more detail testing, simulation and better approaches for testers. This also presents a potential business opportunity in the field of mobile testing tools, test automation and in the increasingly popular cloud-based testing environment.

As previously mentioned, the future trend in ICT is mobility, which also applies for mobile testing. The world of mobile devices is rapidly developing, which requires rapid development in mobile testing.

## REFERENCES

- Abowd, G., Dey, A., & Brown, P. (1999). Towards a better understanding of context and context-awareness.
- Agawi App Glimpse. (n.d.). TouchMarks I: Smartphone Touchscreen Latencies. Retrieved from <http://appglimpse.com/blog/touchmarks-i-smart-phone-touch-screen-latencies/>
- Gao, J., Bai, X., Tsai, W., & Uehara, T. (2013). Mobile Application Testing – Research , Practice , Issues and Needs Testing , Requirements and Features, 4(1).
- Gao, J., Jose, S., Tsai, W., & Uehara, T. (2014). Mobile application testing: A tutorial, 46–55.
- Google. (2014). Supporting Multiple Screens. Retrieved from [http://developer.android.com/guide/practices/screens\\_support.html](http://developer.android.com/guide/practices/screens_support.html)
- Hamblen, M. (2009). Ballmer: We “screwed up with Windows Mobile.” Retrieved from <http://news.idg.no/cw/art.cfm?id=F2F7C35E-1A64-67EA-E4BC04F120F0B898>
- Kirubakaran, B., & Karthikeyani, V. (2013). Mobile Application Testing – Challenges and Solution Approach through Automation.
- Knab, K. (2012). Main issues in mobile app testing. *Testing Experience*, 19.
- Muccini, H. (2012). Software testing of mobile applications: Challenges and future research directions, 29–35.
- Rivera, J., & Van der Meulen, R. (2013). Gartner Says Worldwide PC, Tablet and Mobile Phone Combined Shipments to Reach 2.4 Billion Units in 2013. Retrieved from <http://www.gartner.com/newsroom/id/2408515>
- Schulte, M., & A. Majchrzak, T. (2012). Context-Dependent Testing of Apps Applications. *Testing Experience*.
- Tharakan, M., & Jacob, J. (2012). Roadblocks and their workaround while testing Mobile Applications. *Testing Experience*, 19.
- Yeap, G. (2013). Smart mobile SoCs driving the semiconductor industry: technology trend, challenges and opportunities. *IEDM Technical Digest*, 16–23. Retrieved from <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Smart+Mobile+SoCs+Driving+the+Semiconductor+Industry+:+Technology+Trend+,+Challenges+and+Opportunities#0>