UDC 621.38

O. STAROV, S. VILKOMIR

East Carolina University, NC, USA

CLOUD SERVICES AND TOOLS FOR MOBILE TESTING

Mobile application testing and testing over a cloud are two highly topical fields. This paper provides a state-of-the-art review of cloud-based mobile testing. Mobile testing presents specific test activities, including verification of an application against a variety of heterogeneous smartphone models and versions of operating system, build distribution and test team management, monitoring and user experience analytics of an application in production, etc. Cloud benefits are used to support mobile testing and different cloud services exist to provide remote mobile device hosting, automate tests, deliver builds to testers, and gather statistics of an app's usage. Such cloud services and tools are reviewed in this paper. The analysis of how existing functionalities cover all mobile testing needs is provided, and the lack of integrated testing techniques like combinatorial coverage strategies, test generation services, and services for automated dynamic security testing is observed.

Keywords*: mobile testing, cloud testing, device cloud, application lifecycle management, testing techniques, performance testing, user experience testing, Android.*

Introduction

Mobile development is characterized by a variety of applications with different quality requirements. Online application stores, like the Apple App Store [1] or Google Play [2], offer thousands of market-oriented apps—mobile games, utilities, navigators, social networks, and clients for web resources. At the same time, the interest in critical mobile applications is growing. For instance, online banking has evolved into mobile banking [3], mobile social alerts are widely used to report accidents [4] or warn about hurricanes [5], special apps exist to monitor traffic [6], and help cardiac patients [7]. Augmented reality apps are used for complex navigation and involve a variety of sensors. Mobile applications even support processes at such critical facilities as nuclear power plants [8].

These trends require high levels of reliability and quality for mobile software systems. It affects testing, in particular, and the whole mobile development process in general. Too often, the mobile development process ends with the submission of a social application to an online store. The aim is to gain a wider audience of users in a shorter time, but this does not guarantee the quality of the product and non-critical bugs are usually accepted. Some surveys have confirmed that mobile developers usually deal with small apps and do not adhere to a formal development process [9]. In contrast, a totally different approach is required for critical or business-critical mobile applications, including mobile clients for trustworthy enterprise systems and solutions; for example, Facebook's iOS app is crucial for maintaining the company's profile and reputation and thus was rebuilt to overcome the poor quality of the first version [10].

Advanced mobile software processes typically work according to the Agile-based methodology [11, 12] and include usage of build distribution services to assist in testing, analytical services for maintenance during production, and services to obtain a wider range of mobile devices for testing. These services create a large set of testing-as-a-service (TaaS) resources, or supporting web-applications, that use cloud benefits to facilitate the testing of mobile applications and cover all of the specific mobile testing needs. These cloud solutions make mobile testers more effective because they provide complex infrastructure and/or services that are not feasible within small developer companies.

The goal of this paper is to provide a review of existing cloud services and the tools that support mobile testing and evaluate the coverage of the in-demand functionalities.

This paper is organized as follows. Section 1 provides a brief introduction to cloud testing. Section 2 describes the state-of-the-art mobile testing and different supporting cloud services used during mobile development process. An analysis of the present testing techniques is also provided. Finally, topical research directions in cloud facilities for mobile testing are summarized in the conclusion.

1. Cloud Testing

Many research papers have stated that testing extensively migrates to the cloud nowadays [13–17]. Reviews and classifications of testing cloud services include solutions for web systems and mobile development [18, 19]. Cloud benefits are used not only to support performance, load, or reliability testing of websites, but also to assist with providing required hardware resources (i.e., remote smartphones) for different needs for mobile testing. Cloud-based mobile testing is a young but very topical issue [20].

The database at the Cyber Security and Information Systems Information Analysis Center provides a large list of cloud testing references [21]. Technical and research issues about testing over the cloud are analyzed in [22] and [23] respectively.

This paper uses the term "cloud service" as the most general understanding of cloud computing [24], i.e., cloud service is a software tool or hardware resource that is delivered over the Internet. The definition means that we also take into account such web resources as build distribution solutions and online issue tracking systems. The term "device cloud" (i.e., mobile device cloud or cloud of devices) will also be used, pointing to both the cloud service's nature and the many geographically dispersed devices.

Many specialized studies exist regarding the general architecture and construction of cloud and distributed systems [24, 25], including providing service through application programming interfaces (APIs) [26]. Technical issues for the tests on the cloud are discussed in [23], including Hadoop usage for test distribution. Device clouds require special algorithms for effective test distribution to make overall test execution time as minimal as possible. A comparison of general load balancing algorithms can be found at [27].

2. Mobile Testing

Mobile development has a set of distinctive features and the following specific challenges can be mentioned [9]: support of many hardware and software platforms, correct work with a variety of sensors, interconnections with other applications, high requirements for users' experiences and the quality of the user interface [28, 29], and the existence of web mobile and hybrid applications that incorporate all of these challenges to web development.

Mobile applications are popular among startups and approaches for quick prototyping to evaluate the concept of an app [30] are now in high demand. All of these features contribute to the complexity and specifics of mobile testing [31, 32]. As for mobile testing in this paper, we mean comprehensive testing of a mobile system that includes the testing of mobile apps as well as mobile operation systems (OS) and the related hardware. Different investigations have pointed to the required mobility of the apps in terms of their ability to function in different environments and configurations as the root challenge of testing [31].

uTest published *The Essential Guide to Mobile App Testing* [32], a book that comprehensively and coherently describes challenges and techniques in mobile application testing. A lot of research exists about automation and facilitation of the testing process, including leveraging of cloud abilities [15, 33–38]. Companies that provide cloud services for mobile testing (cloud of devices) usually assist their customers with a set of guides [39, 40].

Examples of testing matrixes to cover all smartphone models or OS versions generate an enormous number of combinations [32]. The issue is significant for the Android platform because of its representatively large number of supported devices with different characteristics (e.g., screen resolution, size of memory, and set of sensors). The problem is compounded by the fact that a smartphone simulator or an emulator cannot fully substitute for the hardware [32]. At the same time, the development for different mobile platforms looks similar. Platforms have similar developer websites with necessary documentation, examples, and suggested patterns [41]. The principles of the application life cycle are similar, for instance, comparing Android to the Windows Phone 7 [42].

Many software development companies are interested in the mobile market and many mobile platforms now exist: Android, iOS, Windows Phone, Symbian, etc. New ones appear regularly like the recent Ubuntu Mobile OS [43]. According to Gartner, Android devices have most of the market [44] and *Forbes* says that the Android platform aims to meet enterprise requirements in the near future [45]. Previous research on the bug statistics for the Android OS [46] proved that the Android (with Symbian) has effectively organized a open-sourced bug-tracking system that deals with bugs and makes the platform better. The number of applications in Google Play is now more than 600,000 and is increasing steadily [47]. The open source nature of Android makes it popular among the scientific community, and many examples of research studies targeted at the Android system can be found.

To facilitate mobile testing, various cloud benefits are used and different TaaS, or supporting services, exist. Figure 1 provides references to them, along with mapping to correspondent testing stages. The presented types of testing were partially taken from a diagram on Perfecto Mobile's guide that shows the demanded device allocation during different application lifecycle management (ALM) stages [39]. The diagram was extended by adding conceptualizations as a separate ALM activity, plus concept, security, and user experience (UX) testing, as well as highlighting test activities such as test planning, management, and issue tracking that are all specific to real-life mobile development.

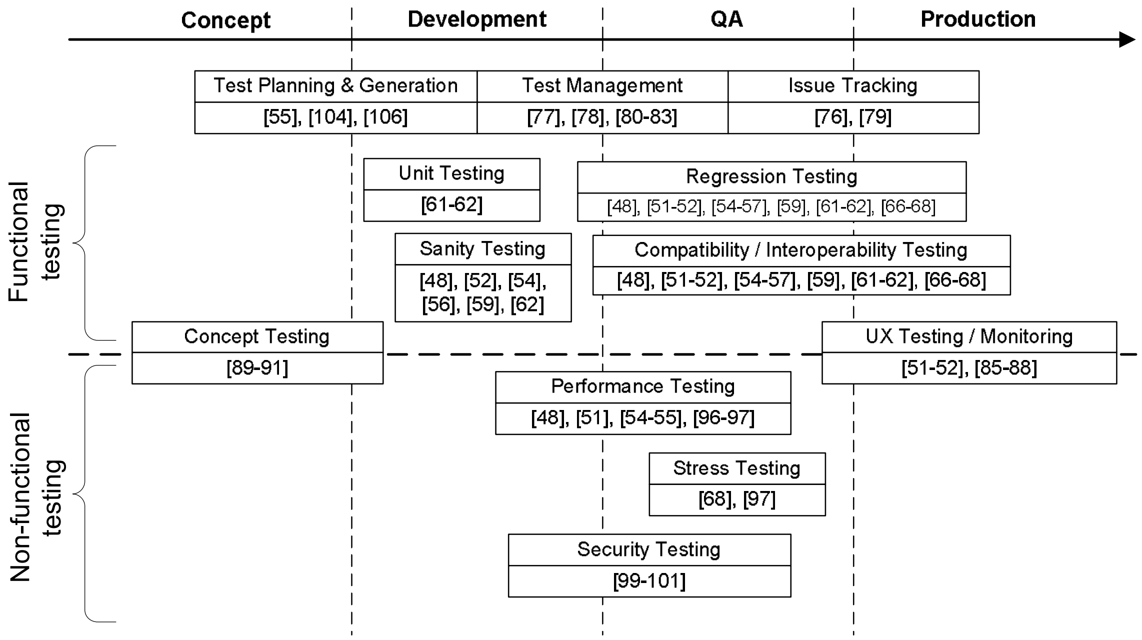

Fig. 1. Test stages and activities with references to correspondent cloud services

The set of cloud services for mobile testing can be divided into three types: device clouds (mobile cloud platforms), services to support ALM, and tools to provide processing according to some testing techniques. The following sub-sections describe each type separately.

2.1. Device Clouds

The majority of cloud services for mobile testing serves as a “cloud of devices” and provides remote access to smartphones in the cloud in order to accomplish testing, in other words, provides device hosting. Such services usually aid mobile developers in using remote smartphones as real devices for manual testing (interactive testing through a web interface), recording of scripts, and automatic running of tests on a range of models. For instance, Perfecto Mobile service [48] provides all of this functionality representing different modern hardware and software mobile platforms (Android, iOS, Windows Phone, and Symbian) and can be integrated with HP UFT (QTP) [49] or MS Team Foundation Server [50]. Devices available in the system have different parameters, for example, testing different types of Internet connections is possible. The service works with two kinds of test scripts: QTP and the Perfecto Mobile Application. Perfecto Mobile is only a public service, but UFT Mobile [51] can also be deployed as a private cloud. UFT Mobile provides automated functional testing and special solutions for realistic mobile performance testing (e.g., LoadRunner and Performance Center).

Keynote DeviceAnywhere [52] is a similar service that provides online manual and automated testing of a mobile app on a variety of devices. It can be integrated with existing ALM through HP QTP, IBM RQM [53] or special Java APIs.

The SOASTA service [54] provides two advanced solutions: TouchTest test automation for multi-touch, gesture-based applications and CloudTest for scalable mobile application testing (performance or load-testing with millions of geographically distributed emulated users). TouchTest scripts can be recorded and performed against user’s own device. Users can control test devices via IP addresses.

The Cigniti device cloud [55] provides remote access to a variety of mobile devices via own proprietary mobile test automation framework, with test accelerators for test automation and performance testing. Cigniti is suitable for network carrier testing.

SeeTest by Experitest [56] provides device cloud that can be deployed as a private platform within an organization. Test automation facilities include test script recording/performing on real devices or emulators and integration with HP UFT (QTP), TestComplete, C#, RFT, Java, Perl, Python. SeeTest also provides manual testing tools.

The CloudMonkey service [57] runs MonkeyTalk scripts across many Android emulators and iOS simulators. Screenshot reports are positioned as the base testing results. CloudMonkey test jobs can be integrated with continuous integration (CI) servers like Jenkins [58].

The Appium on Sauce service [59] covers two functionalities: iOS device hosting and easy CI. The latter means that it can be used as a build server and testers do not need to set up developer environment on local machines. Test automation is implemented with

Selenium [60], and interactive testing is only possible for web mobile applications. Appium can be deployed privately.

The TestDroid Cloud [61] is a device cloud service oriented towards Android apps testing that uses the TestDroid AppCrawler engine to verify application devices' compatibility. TestDroid Recorder can be used to generate reusable Android JUnit test cases. Test results consist of screenshots and device logs. A tester can compare screenshots to check for GUI bugs. TestDroid can also be integrated with Jenkins or leveraged through REST APIs.

The Scirocco Cloud [62] has all of the functionality of a device cloud, except of script recording. It supports only the Android platform and provides manual access to remote devices through its HTML5 web interface. Test automation is done by using one of three drivers: AndroidDriver, MonkeyRunner, or NativeDriver. Results are provided as a set of screenshots to compare.

The LessPainfull device cloud [63] is oriented for Android and iOS apps testing. As a test automation engine, it uses Calabash for Cucumber [64] and accepts Cucumber-based test scripts. LessPainfull provides two options: private cloud tailored for single customer and shared cloud with devices common for several customers.

TestQuest [65] is a distributed framework for deployment within an organization. It is oriented towards Android application testing and can be integrated with MS Visual Studio.

The ZPX service provides device hosting and mobile test automation in the cloud [66] and is compatible with HP ALM products.

Jamo [67] provides a set of tools to perform remote and scheduled testing on a device. For instance, Wanconnector in combination with Remote Device Screen provides access to a device within different geographical locations. The M-eux Test tool supports web application testing.

Apkudo's device analytics [68] provide some elements of multidirectional testing by testing devices (e.g., new smartphone models) against the top 200 apps from the market [69]. Similar services are available for smartphone hardware testing, but these have no relation to mobile apps like Datum [70] that provides verification of calls, data quality, and video quality. Apkudo also offers free public and fully automated stress testing of the Android applications on the big range of models using the Monkey tool [71].

Table 1 summarizes the device clouds mentioned above and a comparison based on supported mobile platforms, types of testing, and delivery type of cloud solution. Manual testing means the remote operation of a device via a web interface, and automated testing incorporates functional and regression testing and different kinds of automation. All device clouds provide compatibility testing as intended. Public cloud means service with shared devices, while a private cloud means an infrastructure allocated to a single user or a system to be deployed on a user-developer's site.

Two known research attempts within universities to create and investigate test-bed cloud solutions for mobile development are SmartLab [72] and the Android Tactical Application Assessment and Knowledge (ATAACK) Cloud [73]. Both are distributed systems that connect a set of mobile devices under the Android OS for application investigation, development, and testing.

The SmartLab is an experimental test-bed being developed at the University of Cyprus.

It provides more than 40 connected Android smartphones plus emulated devices, but not many details are described or known.

Table 1

Device Clouds

Cloud Service	Supported Platforms			Types of Testing	Delivery Type	
	Android	iOS	Other		Public	Private
Apkudo [68]	+			Stress (automated), New device approval	+	
Appium on Sauce [59]		+		Manual for web applications, Automated	+	+
Cigniti [55]	+	+	+	Automated, Interoperability, Performance, Network	+	
CloudMonkey [57]	+	+		Automated, UI-oriented	+	+
DeviceAnywhere [52]	+	+	+	Manual, Automated, Monitoring, Coverage	+	+
Jamo [67]	+	+	+	Automated		+
Perfecto Mobile [48]	+	+	+	Manual, Automated, Performance, Monitoring	+	
Scirocco Cloud [62]	+			Manual, Automated	+	
SeeTest [56]	+	+	+	Manual, Automated, On a new devices		+
SOASTA [54]	+	+	+	Manual, Automated, Load, Performance, Gesture-based	+	+
TestDroid Cloud [61]	+			Automated, UI-oriented, On a new devices	+	+
UFT Mobile [51]	+	+	+	Automated, Load, Performance, Monitoring		+
Zap-Fix [66]	+	+	+	Automated		+

The ATAACK Cloud is new joint project for Virginia Tech, the University of Maryland, and Vanderbilt University, with the support and funding by Air Force Research Laboratories. Its goal is large-scale mobile application testing and investigations.

These research studies consider device clouds with several smartphones connected to one computer (vertical) and several computers with connected smartphones (horizontal) scaling of devices, i.e., fully distributed systems, and how to provide access and testing.

Many studies regarding less-scaled test frameworks for distributed mobile testing [74] that are not cloud services and many tools for vertical-scaled test automation only [75] exist, but their reviews are beyond the scope of this paper.

All services mentioned in this section appear in Figure 1 with the following logistics: services that support the running of unit tests listed under "unit testing," services that support online manual testing listed under "sanity testing," references to script automation techniques of these services listed under "regression testing," all cloud devices listed under "interoperability/compatibility testing," and references to special integrated non-functional test approaches of these services as listed under correspondent types of testing (see section 2.3 for examples).

2.2. Services to Support ALM

The application lifecycle management of mobile applications has own specifications and many cloud services exist that support test-related activities within ALM. Several examples of these cloud services are listed below.

1. Mobile developers, like all software developers, use issue tracking systems, e.g., with Agile-oriented plugins [76], more complex solutions like IBM Rational Quality Manager [77], or test management systems like TestRails [78]. Some of these are integrated with software configuration management and facilitate code reviews or code style checks [79]. A review of similar tools and solutions is not the goal of this paper, so Figure 1 shows only several base examples.

2. Mobile testing involves the use of actual hardware and so testers need additional knowledge and skills, such as build installation or crash-log retrieving. To facilitate beta build distribution activities, many cloud services exist [80–83]. Some of them provide functions for test team management [80] or build provisioning and deployment to the store (AirOnApp for iOS [81]). TestFlight service [80] helps to deal with the iOS build management and distributes them via email between separated testers. It provides an easy application installation on a real device, i.e., by a tap on the link in an email opened on a smartphone. A similar service for Android is Launchpad [82]. The HokeyApp [83] build distribution provides extended functionality to collect live crash reports, feedback from users, and analysis of resulting test coverage. Usage of these services for build distribution can be integrated along with the continuous integration process of the company [84] (e.g., via job scripts for the Jenkins build server [60]).

3. User experience testing and monitoring of an app in production are required activities within mobile testing. Several analytics services gather usage statistics like [85] and these can be incorporated in a mobile app. Perfecto Mobile service also provides some solutions for monitoring performance [86].

The following two services incorporate user experience testing in the build distribution facilities. The UserTesting service [87] provides many real users who will examine an app and provide feedback about their experience with the app and thoughts about it. The Amazon A/B testing for Android [88] provides a service that distributes two builds that differ in some features between two unique groups of users. Then it provides measurements and results about which feature is more successful.

4. Mobile development is very popular among startups and usually requires rapid prototyping for concept feasibility evaluation. Thus such services exist like [89] to easily create interactive prototypes, or [90] to share an app demo, or [91] to create realistic mockups. All of these are needed to test the concept and idea of the app (i.e., if it can hit the market) at a minimal expense.

2.3. Testing Techniques Provided

This section discusses testing techniques on existing cloud services for mobile development. Device clouds described in section 2.1 provide different techniques for test automation (recording, distribution, and execution). This includes unit tests and GUI-based testing. Examples of approaches are standard Android SDK tools MonkeyRunner [92] and Monkey [71], special solutions like SOASTA TouchTest, and solutions based on object recognition (e.g., Eggplant automation based on VNC technology [93]).

Test automation has its own weak sides, and according to experts in the field, cannot serve as a total substitution for manual testing [94]. The issue we noticed during the analysis of cloud test automation was the delivery of the test input data to mobile sensors (GPS, accelerometer, camera, etc.). While solutions to send dummy GPS coordinates exist, situation with a photo camera is more complicated because it requires the simultaneous changing of a picture (preferable physically in front of a camera) while performing a script. A variety of mobile apps use a camera as a part of their key functionality (e.g., shopping apps and QR code readers [95]), and proper testing requires test cases with snapshots from different distances, angles, lights, etc. Other problematic aspects of automation are the sophisticated (approximate) screenshots comparisons, executions of direct device-to-device communication during the test, etc.

Device clouds provide compatibility, interoperability, and regression testing. Many services provide embedded tools to support performance monitoring and load testing [48, 51, 54, 55] or even automated stress testing on a variety of devices [68].

There are special cloud services that aid with mobile performance and load testing. For instance, SandStrom [96] can be used for load testing of web mobile applications and NeoLoad [97] focuses on load testing of back-end servers by emulating typical mobile devices working in parallel and sending appropriate content to the server. There are also standalone solutions for test techniques applications like performance frame counters on Windows Phone Emulator [98] that theoretically can be leveraged in a cloud.

Security testing is mainly presented by static check techniques. Checkmarks [99] provides scanning of source code and supports Android and iOS applications. Mobile App Security and Privacy Analysis by Veracode [100] scans and evaluates binary files for vulnerabilities and can be leveraged through APIs. Another type of services exists based on experts. For instance, uTest experts will assist with mobile security testing by manual penetration and using internal static and dynamic security testing solutions [101]. Other solutions to guarantee mobile security focus on proper development processes according to secured methodologies and approaches [102]. At the same time, research papers about novelty mobile security testing approaches exist (that potentially can be leveraged by some cloud services) [103], but these are not described in the present paper.

Concept testing, UX testing, and monitoring techniques were comprehensively described in section 2.2 as parts of services that support ALM.

Mobile testing services should incorporate test planning and test generation techniques. This paper indicates the lack of such functionality. Only Keynote DeviceAnywhere Test Planner [104] provides a coverage calculation for smartphone models to test that can be considered as application of combinatorial testing techniques, but it can be extended by using pairwise [105], t-way [106], or other approaches. HokeyApp only provides test coverage monitoring and analytics, i.e., the matrix of the devices and languages that were tested. Cigniti Test Advisory Services and TestRails provide more high-level test planning and control facilities.

Conclusion

This paper discussed state-of-the-art, cloud-based mobile testing. Existing cloud services and tools to perform or support this testing were listed and categorized. Existing solutions were compared with activities and types of testing required during mobile ALM.

The lack of integrated testing techniques like combinatorial testing for coverage calculation, test generation services, and services for automate dynamic security testing was noted.

Technical challenges in test automation within device clouds were mentioned, highlighting the issue of mobile sensors' control during test execution.

The situation with cloud services for mobile testing is changing extremely rapidly: new ones appear and old ones get new functionalities. Thus, it is hard to guarantee that the provided list of tools and services is exhaustive, but it can serve as a useful baseline.

References

1. Apple App Store [Electronic resourse]. – Access to: https://itunes.apple.com/us/genre/ios/id36?mt=8. – 9.04.2013.

2. Google Play (Android Market) [Electronic resourse]. – Access to: https://play.google.com/store/apps. – 9.04.2013.

3. Mobile Banking [Electronic resourse] / Bank of America. – Access to: https://www.bankofamerica.com /online-banking/mobile.go. – 9.04.2013.

4. White, J. R&D Challenges and Solutions for Mobile Cyber-Physical Applications and Supporting Internet Services [Text] / J. White, S. Clarke, B. Dougherty, C. Thompson, D. Schmidt // Springer Journal of Internet Services and Applications. – November 2010. – V. 1, No. 1 – P. 45 - 56.

5. Carr, D.F. Hurricane Sandy: Mobile, Social Tools Help Emergency Management [Electronic resource] / D.F. Carr // Brainyard. – Access to: http://www. informationweek.com/thebrainyard/news/social_media_ monitoring/240012463/hurricane-sandy-mobile-social-tools-help-emergency-management. – 9.04.2013.

6. Work, D.B. Impacts of the Mobile Internet on Transportation Cyberphysical Systems: Traffic Monitoring using Smartphones [Text] / D.B. Work, A.M. Bayen // Proceedings of National Workshop for Research on High-Confidence Transportation Cyber-Physical Systems: Automotive, Aviation and Rail. – Washington, DC, USA, November 18 - 20, 2008.

7. Leijdekkers, P. Personal Heart Monitoring and Rehabilitation System using Smart Phones [Text] / P. Leijdekkers, V. Gay // Proceedings of International Confer-

ence on Mobile Business (ICMB 2006). – Copenhagen, Denmark, June 26 - 27, 2006. – P. 29.

8. Moser, K. Improving Work Processes for Nuclear Plants [Graphic] / K. Moser // Proceedings of American Nuclear Society Utility Working Conference. – Hollywood, Florida, USA, August 5 - 8, 2012.

9. Wasserman, A. Software engineering issues for mobile application development [Text] / A. Wasserman // Proceedings of the Workshop on Future of Software Engineering Research at the 18th International Symposium on Foundations of Software Engineering (ACM SIGSOFT). – Santa Fe, USA, November 7 - 11, 2010. – P. 397 - 400.

10. Warren, C. Facebook Releases New, Faster iOS App [Electronic resourse] / C. Warren. – Access to: http://mashable.com/2012/08/23/facebook-ios-5/. – 9.04.2013.

11. Holler, R. Mobile Application Development: A Natural Fit with Agile Methodologies [Electronic resource] / R. Holler // VersionOne. – Access to: http://www.versionone.com/pdf/mobiledevelopment.pdf. – 9.04.2013.

12. Denman, J. ALM expert says Agile methods are the way to conquer mobile ALM [Electronic resource] / J. Denman // TheServerSide.com – Access to: http://www.theserverside.com/tip/ALM-expert-says-Agile-methods-are-the-way-to-conquer-mobile-ALM. – 9.04.2013.

13. Vilkomir, S. Cloud Testing: A State-of-the-Art Review [Text] / S. Vilkomir // Information & Security: An International Journal. – 2012. – V. 28, Is. 2, No. 17 – P. 213 - 222.

14. Tilley. S. Software Testing in the Cloud: Perspectives on an Emerging Discipline [Text] / S. Tilley, T. Parveen // IGI Global. – November 2012.

15. Tsai, W. Testing as a service over cloud [Text] / W. Tsai, X. Chen, L. Liu, Y. Zhao, L. Tang, W. Zhao // Proceedings of the 5th IEEE International Symposium on Service Oriented System Engineering. – 2010. – P. 181 – 188.

16. Kalliosaari, L. Testing in the Cloud: Exploring the Practice [Text] / L. Kalliosaari, O. Taipale, K. Smolander // IEEE Software. – 2012. – V. 29, Is. 2 – P. 46 - 51.

17. Weidong, F. Cloud testing: The next generation test technology [Text] / F. Weidong, X. Yong // Proceedings of the 10th International Conference Electronic Measurement & Instruments (ICEMI). – Chengdu, China, August 16 - 19, 2011. – P. 291 - 295.

18. Inçki, K. A Survey of Software Testing in the Cloud [Text] / K. Inçki, I. Ari, H. Soze // Proceedings of the IEEE Sixth International Conference on Software Security and Reliability Companion. – 2012. – P. 18 - 23.

19. Priyanka. Empirical evaluation of cloud-based testing techniques: a systematic review [Text] / Priyanka, I. Chana, A. Rana // ACM SIGSOFT Software Engineering Notes archive. – May 2012. – V. 37, Is. 3 – P. 1 - 9.

20. Mote, D. Cloud based Testing Mobile Apps [Graphic] / D. Mote // 2nd IndicThreads.com Conference On Software Quality. – Pune, India, March 25 - 26, 2011.

21. Cloud Testing [Electronic resourse] / Database of Cyber Security and Information Systems Information Analysis Center. – Access to: https://sw.thecsiac.com/databases/url/key/7848/8764/8765#.USGPb-h8vDm. – 9.04.2013.

22. Tilley. S. Software Testing in the Cloud: Migration & Execution [Text] / S. Tilley, T. Parveen // Springer Briefs in Computer Science. – September 2012.

23. Riungu, L.M., Research Issues for Software Testing in the Cloud [Text] / L.M. Riungu, O. Taipale, K. Smolander // Proceedings of the IEEE 2nd International Conference Cloud Computing Technology and Science (CloudCom). – 2010. – P. 557 - 564.

24. Rhoton, J. Cloud Computing Architected: Solution Design Handbook / J. Rhoton, R. Haukioja // Recursive, Limited. – May 2011.

25. Coulouris, G. Distributed Systems: Concepts and Design / G. Coulouris, J. Dollimore, T. Kindberg, G. Blair // Addison-Wesley. – May 2011 (5th ed.).

26. Jacobson, D. APIs: A Strategy Guide. / D. Jacobson, G. Brail, D. Woods // O'Reilly Media, Inc. – December 2011. – P. 60 - 70.

27. Randles, M. Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing [Text] / M. Randles, D. Lamb, A. Taleb-Bendiab // Advanced Information Networking and Applications Workshops (WAINA), IEEE 24th International Conference. – 2010.

28. User Human Interface Guidelines for iOS [Electronic resourse] / Android Developers. – Access to: http://developer.android.com/design/get-started/principles.html. – 9.04.2013.

29. Design Principles for better User Experience [Electronic resourse] / Android Developers. – Access to: http://developer.android.com/design/get-started/principles.html. – 9.04.2013.

30. Androgen (for Android app generator) [Electronic resourse] / Pretotyping. – Access to: http://www.pretotyping.org/androgen. – 9.04.2013.

31. Muccini, H. Software testing of mobile applications: Challenges and future research directions [Text] / H. Muccini, A. Di Francesco, P. Esposito // Proceedings of the 7th International Workshop on Automation of Software Test (AST). – 2012.

32. The Essential Guide to Mobile App Testing (free eBook) [Electronic resourse] / uTest, Inc. – Access to: http://www.utest.com/landing-blog/essential-guide-mobile-app-testing. – 9.04.2013.

33. Franke, D. Providing a Software Quality Framework for Testing of Mobile Applications [Text] / D. Franke, C. Weise // Proceedings of the IEEE 4th International Conference on Software Testing, Verification and Validation (ICST). – Berlin, Germany, March 21 - 25, 2011. – P. 431 - 434.

34. Milano D. Android Application Testing Guide [Text] / D. Milano // Packt Publishing Ltd. – June 2011.

35. Frederick G. Testing a Mobile Web Site [Text] / G. Frederick, R. Lal // Beginning Smartphone App Development, Part IV, Apress. – 2009. – P. 259 – 272.

36. Sama, M. Context-Aware Adaptive Applications: Fault Patterns and Their Automated Identification [Text] / M. Sama, S. Elbaum, F. Raimondi, D. Rosenblam, Z. Wang // IEEE Transactions on Software Engineering. – September - October 2010. – P. 644 - 661.

37. Dantas, V. Testing requirements for mobile applications [Text] / V. Dantas, F. Marinho, A. Da Costa, R. Andrade // Proceedings of the 24th International Symposium on Computer and Information Sciences (ISCIS). – 2009.

38. Test Strategies for Smartphones and Mobile Devices [Electronic resource] / Macadamian – Access to: http://www.macadamian.com/images/uploads/whitepapers/MobileTestStrategies_Aug2010.pdf. – 9.04.2013.

39. Make Your Mobile Testing Solution Enterprise-Ready [Electronic resourse] / Perfecto Mobile. Access to: http://www.perfectomobile.com/portal/cms/resources/enterprise-ready_white-paper. – 9.04.2013.

40. Testing Strategies and Tactics for Mobile Applications [Electronic resourse] / Keynote. – Access to: http://www.keynote.com/docs/whitepapers/WP_Testing_Strategies.pdf. – 9.04.2013.

41. Android Developers [Electronic resourse]. – Access to: http://developer.android.com. – 9.04.2013.

42. Franke D. Reverse Engineering and Testing Service Life Cycles of Mobile Platforms [Text] / D. Franke, C. Elsemann, S. Kowalewski // Proceedings of the 23rd International Workshop on Database and Expert Systems Applications (DEXA). – 2012. – P. 16 - 20.

43. Suri, I. Canonical Unveils the Ubuntu Phone at CES 2013 [Electronic resourse] / I. Suri. – Access to: http://siliconangle.com/blog/2013/01/10/canonical-unveils-the-ubuntu-phone-at-ces-2013/. – 9.04.2013.

44. Haselton, T. Android Has 56.1% Of Global OS Market Share, Gartner Says [Electronic resourse] / T. Haselton // TechnoBuffalo. – Access to: http://www.technobuffalo.com/2012/05/16/android-has-56-1-of-global-os-market-share-gartner-says. – 9.04.2013.

45. Fidelman, M. The Latest Infographics: Mobile Business Statistics For 2012 [Electronic resourse] / M. Fidelman // Forbes. – Access to: http://www.forbes.com/sites/markfidelman/2012/05/02/the-latest-infographics-mobile-business-statistics-for-2012. – 9.04.2013.

46. Maji, K. Characterizing Failures in Mobile OSes: A Case Study with Android and Symbian [Text] / K. Maji, K. Hao, S. Sultana, S. Bagchi // Proceedings of the 21st International Symposium in Software Reliability Engineering (ISSRE). – 2010. – P. 249 - 258.

47. Android Statistics [Electronic resourse] / AppBrain. – Access to: http://www.appbrain.com/stats/. – 9.04.2013.

48. The MobileCloud Company [Electronic resourse] / Perfecto Mobile. – Access to: http://www. perfectomobile.com. – 9.04.2013.

49. UFT Mobile, the official HP mobile testing solution [Electronic resourse] / Perfecto Mobile. – Access to: http://www.perfectomobile.com/portal/cms/services/uft_mobile. – 9.04.2013.

50. Sterling, C. Testing devices made easier with Perfecto Mobile's MobileCloud platform and Team Foundation Server [Electronic resourse] / C. Sterling. // MSDN Blogs. – Access to: http://blogs.msdn.com/b/visual studioalm/archive/2013/02/21/testing-devices-made-easier-with-perfecto-mobile-s-mobilecloud-platform-and-team-foundation-server.aspx. – 9.04.2013.

51. Whiting, R. New HP Tools Help Developers Test Cloud, Mobile Applications [Electronic resourse] / CRN. – Access to: http://www.crn.com/news/applications-os/240049875/new-hp-tools-help-developers-test-cloud-mobile-applications.htm – 9.04.2013.

52. Keynote DeviceAnywhere [Electronic resourse] / Keynote. – Access to: http://www.keynotedeviceanywhere.com. – 9.04.2013.

53. Galde, J. Keynote's DeviceAnywhere platform Fully Integrated with RQM 3.0.1 [Electronic resourse] / J. Galde // IBM developerWorks. – Access to: https://www.ibm.com/developerworks/mydeveloperworks/blogs/deviceanywhere/entry/keynote_s_deviceanywhere_platform_fully_integrated_with_rqm_3_0_12?lang=en. – 9.04.2013.

54. CloudTest On-Demand [Electronic resourse] / SOASTA. – Access to: http://www.soasta.com/services/cloudtest-on-demand. – 9.04.2013.

55. Cigniti Mobile Testing [Electronic resourse] / Cigniti. – Access to: http://www.cigniti.com/mobile-testing. – 9.04.2013.

56. SeeTest Mobile Cloud [Electronic resourse] / SeeTest. – Access to: http://experitest.com/cloud/. – 9.04.2013.

57. CloudMonkey *[Electronic resourse] / Gorilla Logic. – Access to: https://www.gorillalogic.com/ cloud-monkey. – 9.04.2013.*

58. Appium on Sauce [Electronic resourse] / Sauce Labs. – Access to: https://saucelabs.com/appium. – 9.04.2013.

59. Selenium RC [Electronic resourse] / SeleniumHQ. – Access to: http://docs.seleniumhq.org/ projects/remote-control/. – 9.04.2013.

60. Jenkins Continuous Integration [Electronic resourse] / Jenkins. – Access to: http://jenkins-ci.org. – 9.04.2013.

61. Testdroid Cloud [Electronic resourse] / BitBar. – Access to: http://testdroid.com/product/testdroid-cloud. – 9.04.2013.

62. Scirocco Cloud [Electronic resourse] / Scirocco. – Access to: http://www.scirocco-cloud.com/en/. – 9.04.2013.

63. LessPainful [Electronic resourse] / LessPainful. – Access to: https://www.lesspainful.com. – 9.04.2013.

64. Calabash-Android [Electronic resourse]. – Access to: https://github.com/calabash/calabash-android. – 9.04.2013.

65. TestQuest 10 [Electronic resourse] / Bsquare. – Access to: http://www.bsquare.com/products/testquest-automated-testing-platform/testquest-10. – 9.04.2013.

66. ZPX for Mobile [Electronic resourse] / Zap-Fix. – Access to: http://www.zaptechnologies.com/products/zap-fix/portfolio/zap-fix-for-mobile. – 9.04.2013.

67. Jamo Solutions Technology [Electronic resourse] / Jamo Solutions. – Access to: http://www.jamosolutions.com/technology.html. – 9.04.2013.

68. Apkudo [Electronic resourse]. – Access to: http://www.apkudo.com. – 9.04.2013.

69. Rowinski D. Mobile Carriers and OEMs Get Android App Testing Cloud from Apkudo [Electronic resourse] / D. Rowinski // ReadWrite. – Access to: http://readwrite.com/2012/02/07/mobile-carriers-and-oems-get-a. – 9.04.2013.

70. Datum [Electronic resourse]. – Access to: http://www.metricowireless.com/systems/measdata.cfm. – 9.04.2013.

71. UI/Application Exerciser Monkey tool [Electronic resourse] / Android Developers. – Access to: http://developer.android.com/tools/help/monkey.html. – 9.04.2013.

72. Konstantinidis, A. Demo: a programming cloud of smartphones [Text] / A. Konstantinidis, C. Costa, G. Larkou, D. Zeinalipour-Yazti // Proceedings of the 10th international conference on Mobile systems, applications, and services. – 2012. – P. 465 - 466.

73. Turner, H. Building a Cloud-Based Mobile Application Testbed [Text] / H. Turner, J. White, J. Reed, J. Galindo, A. Porter, M. Marathe, A. Vullikanti, A. Gokhale // IGI Global. – November 2012. – P. 382 - 403.

74. She, S. Hermes: A Tool for Testing Mobile Device Applications [Text] / S. She, S. Sivapalan, I. Warren // Proceedings of the Software Engineering Conference (ASWEC). – Queensland, Australia, April 14 – 17, 2009.

75. User Scenario Testing for android [Electronic resourSe] / Robotium. – Access to: http://code.google.com/p/robotium/. – 9.04.2013.

76. YouTrack [Electronic resourse] / JetBrains. – Access to: http://www.jetbrains.com/youtrack. – 9.04.2013.

77. IBM RQM [Electronic resourse]. Access to: http://www-142.ibm.com/software/products/us/en/ratiqual mana. – 9.04.2013.

78. TestRails [Electronic resourse]. / GurockSoftware. – Access to: http://www.gurock.com/testrail/. – 9.04.2013.

79. Assembla [Electronic resourse]. – Access to: https://www.assembla.com. – 9.04.2013.

80. TestFlight [Electronic resourse]. – Access to: https://testflightapp.com. – 9.04.2013.

81. Air On App [Electronic resourse]. – Access to: http://www.aironapp.com.

82. Launchpad [Electronic resourse]. – Access to: http://launchpadapp.com. – 9.04.2013.

83. HokeyApp [Electronic resourse]. – Access to: http://hockeyapp.net. – 9.04.2013.

84. Flower M. Continuous Integration [Electronic resource] / M. Flower. – Access to: http://martinfowler.com/articles/continuousIntegration.html. – 9.04.2013.

85. Google Analytics [Electronic resource] / Google. – http://www.google.com/analytics. – 9.04.2013.

86. MobileCloud Monitoring [Electronic resource] / PerfectoMobile. – Access to: http://www.perfectomobile.com/portal/cms/services/monitoring.html. – 9.04.2013.

87. User Testing [Electronic resourse]. – Access to: http://www.usertesting.com/mobile. . – 9.04.2013.

88. Amazon A/B testing for Android [Electronic resourse] / Amazon Mobile Build Distribution. – Access to: https://developer.amazon.com/sdk/ab-testing/reference/how-ab-works.html. – 9.04.2013.

89. FluidUI [Electronic resourse]. – Access to: https://www.fluidui.com. – 9.04.2013.

90. Kickfolio [Electronic resourse]. – Access to: http://kickfolio.com. – 9.04.2013.

91. Pidoco [Electronic resourse]. – Access to: https://pidoco.com/en/tour/why-pidoco. – 9.04.2013.

92. MonkeyRunner [Electronic resourse] / Android Developers. – Access to: http://developer.android.com/tools/help/monkeyrunner_concepts.html. – 9.04.2013.

93. EggPlant for Mobile [Electronic resourse] / TestPlant. – Access to: http://www.testplant.com/ products/eggplant/mobile/. – 9.04.2013.

94. Whittaker, J. How Google Tests Software [Text] / J. Whittaker, J. Arbon, J. Carollo // Addison-Wesley Professional. – April 2012.

95. Nextag Shopping [Electronic resourse] / Nextag Mobile. – Access to: https://play.google.com/store/apps/details?id=com.nextag.android&hl=en. – 9.04.2013.

96. SandStrom [Electronic resourse]. – Access to: http://sandstorm.impetus.com/. – 9.04.2013.

97. NeoLad [Electronic resourse]. – Access to: http://www.neotys.com/product/mobile-load-testing.html. – 9.04.2013.

98. Test your app's performance [Electronic resourse] / Windows Phone Dev Center. – Access to: http://msdn.microsoft.com/library/windowsphone/develop/jj247547(v=vs.105).aspx#BKMK_testperf. – 9.04.2013.

99. Checkmarks [Electronic resourse]. – Access to: http://www.checkmarx.com. – 9.04.2013.

100. Veracode [Electronic resourse]. – Access to: http://www.veracode.com/products/mobile-application-security.html. – 9.04.2013.

101. Mobile Security Testing [Electronic resourse] / uTest. – Access to: http://www.utest.com/mobile-security-testing. – 9.04.2013.

102. Security Tips [Electronic resourse] / Android Developer. – Access to: http://developer.android.com/training/articles/security-tips.html. – 9.04.2013.

103. Mahmood, R. A whitebox approach for automated security testing of Android applications on the cloud [Text] / R. Mahmood, N. Esfahani, T. Kacem, N. Mirzaei, S. Malek, A. Stavrou // Proceedings of the 7th International Workshop on Automation of Software Test. – 2012. – P. 22 - 28.

104. Keynote DeviceAnywhere Test Planner [Electronic resourse] / Keynote. – Access to: http://www. keynotedeviceanywhere.com/mobile-test-planner.html. – 9.04.2013.

105. Kuhn, D.R. Practical Combinatorial Testing [Text] / D.R. Kuhn, R.N. Kacker, Y. Lei // NIST Special Publication. – October 2010. – P. 13 - 15.

106. Hexawise [Electronic resourse]. – Access to: http://hexawise.com. – 9.04.2013.

Поступила в редакцию 14.02.2013, рассмотрена на редколлегии 13.03.2013

Рецензент: д-р техн. наук, проф. В.С. Харченко, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, Украина.

ХМАРНІ СЕРВІСИ ТА ІНСТРУМЕНТИ ДЛЯ МОБІЛЬНОГО ТЕСТУВАННЯ

О. Старов, С. Вілкомір

Тестування мобільних програм та хмарне тестування – два актуальні питання сьогодення. Мобільне тестування має певну специфіку, передбачаючи обов’язкову верифікацію програми на багатьох різнорідних моделях смартфонів та версіях операційних систем, організацію ефективної розсилки збірок програм та координацію роботи тестувальників, моніторинг та аналітику досвіду кінцевого користувача. Переваги хмарних обчислень широко використовують для підтримки мобільного тестування, і нині існує безліч хмарних сервісів для дистанційного хостингу смартфонів, автоматизації тестів, доставки версій програми й збирання різноманітної статистики. Такі сервіси й інструменти розглянуто в цій статті. Наведено аналіз покриття всіх запитаних функцій мобільного тестування й спостережено нестачу вбудованих у сервіси методів тестування (зокрема, комбінаторних стратегій покриття), методів генерації тестів і сервісів для динамічного тестування безпеки й захищеності програм.

Ключові слова: мобільне тестування, хмарне тестування, хмара девайсів, хостинг смартфонів, методи тестування, навантажувальне тестування, тестування досвіду користувача, Android.

ОБЛАЧНЫЕ СЕРВИСЫ И ИНСТРУМЕНТЫ ДЛЯ МОБИЛЬНОГО ТЕСТИРОВАНИЯ

А. Старов, С. Вилкомир

Тестирование мобильных приложений и облачное тестирование – два актуальных направления на сегодняшний день. Мобильное тестирование имеет определенную специфику, предполагая необходимую верификацию приложения на множестве разнообразных моделей смартфонов и версий операционных систем, организацию эффективной рассылки сборок приложения и координацию работы тестировщиков, мониторинг и аналитику опыта конечного пользователя. Преимущества облачных вычислений широко используются для поддержки мобильного тестирования, и сейчас существует множество облачных сервисов для удаленного хостинга смартфонов, автоматизации тестов, доставки версий приложения и сбора различной статистики. Такие сервисы и инструменты рассматриваются в данной статье. Приводится анализ покрытия всех востребованных функций мобильного тестирования, и наблюдается нехватка встроенных в сервисы методов тестирования (в частности, комбинаторных стратегий покрытия), методов генерации тестов, и сервисов для динамического тестирования безопасности и защищённости приложений.

Ключевые слова: мобильное тестирование, облачное тестирование, облако девайсов, хостинг смартфонов, методы тестирования, нагрузочное тестирование, тестирование опыта пользователя, Android.

Старов Алексей – студент-магистр, отделение компьютерных наук, Университет Восточной Каролины, США, e-mail: a_starov@hotmail.com.

Вилкомир Сергей – доктор философии, ассоциированный профессор, отделение компьютерных наук, Университет Восточной Каролины, США, e-mail: vilkomirs@ecu.edu.