

PRIMARY RESEARCH

Quality assurance for mobile applications

Abdul Basit ¹, Iftikhar Alam Khan ², Noor Un Nisa ^{3*}

^{1,2} Department of Creative Computing, Bath Spa University, Academic Center RAK, Ras Al-Khaimah, UAE

³ Department of Business Management, Bath Spa University, Academic Center RAK, Ras Al-Khaimah, UAE

Keywords

Quality insurance
Mobile application
Smartphones
Microwave landing
Quality assurance

Received: 04 July 2020

Accepted: 29 September 2020

Published: 16 November 2020

Abstract

The quality of mobile applications is a very important and essential factor for their success among the users otherwise users will reject and discard the applications due to insufficient quality assurance. Mobile applications are becoming popular among the users and due to the peculiarities of mobile applications, there is also a need for quality assurance in the applications to focus and maintain the challenges faced by the users. This paper will identify the main points and expose different approaches which will address the problems and issues of quality assurance for mobile applications. There are research questions and according to those goals are derived. The quality of the mobile application is a very important factor for their success among the users. This depends on the users how they like the applications of the mobiles and successfulness of the mobiles directly depends on the users if the quality assurance for the mobiles for the mobile application should be the best quality. Different popular mobile in the market like I phone, Samsung, Nokia and Motorola are popular mobiles and brands model among the people because of their quality assurance for mobiles applications. FIT4 apps are represented by which quality assurance can be accessed for the mobile's applications for quality purposes. If the quality level is low the brands and models of the mobile are straight way discarded by the users. Two steps empirical evaluation is done for the quality assurance purpose. The basic purpose of the experiment and a case study about the quality assurance of the mobile is performed as post mortem analysis. All mobiles' applications require quality assurance approaches due to context approaches awareness and platform diversity. Some characteristics need to be addressed for quality assurance. There are some budgets restrictions which are sometimes become hindrances in software development.

© 2020 The Author(s). Published by TAF Publishing.

I. INTRODUCTION

In this fast-growing world, mobiles, mobile applications, smartphones and other communicational devices related to mobile phones become most popular and usage of mobiles are increased in this new century [1]. There are different operating systems are introduced in mobiles phone technology like IOS, android, windows, blackberry, and Sym-bian. But people like most smartphones have android systems and apple I Phone. But there are a lot of challenges for the manufactures of hardware and software development for the mobiles to stay in completion otherwise they will be out from the race of technology [2]. There is also competition among the application developers of mobiles and mobile apps.

Automated testing is done with the help of an automated testing tool. The automated testing tool provides computer-

controlled testing rather than manually. The testing tool executes the test cases to test the performance and functionality of the software under test. Automated testing aims to reduce the required human effort as in manual testing but it does not remove the need for manual testing at all [1]. Mobile platforms are being adopted worldwide because of a variety of software being offered to users in those handheld and portable devices. Testing is being used as a quality assurance technique for mobile apps too [2].

Several tools are proposed and implemented for this purpose. These tools have already been evaluated and compared for their unique features, supported platforms, code coverage, and efficiency. However, existing automated testing tools of mobile applications have not been evaluated and compared for different quality attributes they can enhance in apps under test. Therefore, two research objectives are

*Corresponding author: Noor Un Nisa

†email: noor@bathspa.ae



formulated for this study are 1) to evaluate different testing tools of mobile apps focusing on identifying quality factors they aid to achieve in the apps under test; 2) to measure overall trends of essential quality factors achieved in the mobile apps under test using automated testing tools. In this paper, we have evaluated and compared automated testing tools for adding or enhancing valuable quality factors in mobile applications under test.

So, the companies want long term success in the field of mobile phone technology. Mobile phones hardware and software along with different applications which are built-in mobile phones are all tested over several different platforms before launching to the global platform. Functional testing of the mobile's phones is done along with no function testing like security testing and usability testing [3]. The basic purpose of mobile phone testing is to improve the mobile phone hardware, software, and applications along with apps. In this research project, we are going to test the mobile applications along with their competence in mobile testing technology [4]. We can also learn mobile automation which is the main challenge nowadays. Many mobile phone applications testers are confused in understanding the hybrid mobile apps of the native companies. In the market different types of automation tools are available but different tools have their advantage. So that's very easy for the developers to select the best tool for the applications development. The quality of mobile applications for business use and especially for the company stakeholders can be highly trusted and if the application failure has happened it can take lead to serious failure and consequences. Mobiles applications become popular among people and these applications in games, medicine, science and on different topics are launched. The mobile application in insurance quality proposes approaches to support inspections [5].

A. Background

The success of any software product is determined by the quality of that software. This gives software quality assurance a great opportunity in the software industry and customer satisfaction drives it. To develop a product of good quality and without any defects within the cost and time constraints have become critical. Implementing such products, with minimum or no bugs is a difficult task. This is the reason that the concept of software testing has got its existence [3]. In the software industry, the testing of software has become an extensive and vital phase of SDLC. It also provides a final evaluation of other activities such as requirements specification, software design, and coding [4]. Software testing is an activity, which is performed to eval-

uate the correctness and functionality of the software for assuring the fulfilment of user requirements and expected quality [6]. IEEE defines software testing as the process to evaluate the system or its components manually or by automated means to determine whether it fulfils the user requirements or to find the difference between actual result and expected result [7]. Hence, software testing is to execute software to identify defects or any missing features that were expected by the user requirements. Software testing results in improved quality and effectiveness of the software system if it is executed appropriately. Detecting the defects in software and removing those defects before the release of software leads to reduced maintenance costs.

B. Objectives

1. Comparative Analysis of Software Testing Techniques for Mobile Applications
2. An Approach for Evaluating and Improving the Test Processes of Mobile Application Developments with manual method
3. Quality assurance of mobile applications: a systematic mapping study

C. Software Testing

Normally software testing is considered as an activity for the detection of defects whereas there are different reasons behind the conduction of software testing. Improved software quality is one of the major reasons. Software quality is improved by ensuring that the software product fulfils the user requirements and expectations. The smooth functioning of the software system can be ensured through testing. The software developing industries spend most of their time and cost on software testing during the SDLC [8]. If the testing is done early in the SDLC to prevent the occurrence of defects, it reduces the time and cost spent whereas, if the defects are detected in later stages, then the time to market and cost rises significantly. Therefore, performing testing throughout the SDLC is a better practice to detect the defects of the software. It is less expensive to remove the defects earlier, even before the release of the software [9]. Software testing aims to evaluate the capabilities of an application or the software and verify that it fulfils the quality principles such as reliability, portability, efficiency, security, usability, etc. Through testing, all these principles should also be verified and ensured [5]. There are two main objectives of software testing. First, the detection of errors or defects. Second, preventing the number of occurrences of defects in the software system results in overall improved efficiency of the system.

B. Manual Software Testing

Manual testing is the simplest level of testing in which the tests are executed as per test cases and by directly interacting with the software. In this testing, the tester prepares the test cases. Test cases, are the explanations of the features and the expected results of the software under test and are written in simple natural language. The process of manual testing becomes too much time-taking as it requires all the activities to be performed manually. Though, manual testing is preferred in the case of some complex systems where a few critical defects can only be discovered while testing manually. During manual testing, the tester interacts with the system under test as the end-user of that software would, and ensures the effectiveness of the system by navigating through the software [10]. Manual testing has the following drawbacks [10]:

1. Time-taking
 2. Requires more testers
 3. Less accurate results
 4. Testing multiple features in parallel, not possible
- Lack of reusability of tests

II. METHODOLOGY

A. Work and Quality Assurance of Mobile Applications

Quality assurance of mobile apps deals with high-quality apps is produced based on standards and deliver high-quality products line for the consumers [11]. The mobile apps which are not according to standards and high quality are immediately removed from the mobiles and uninstall. Nowadays banks and all other businesses are run on a lie

and all types of work of business are done on mobile phones and customized apps [10]. Users can use several different types of apps which are helpful in businesses to get immediate responses from these applications and make users records from these applications. Especially, the bank's application is designed according to customers need and customers can do a lot of transactions by using these apps and they do not need to go to the banks [11]. With the help of these applications, they can sort out their problems quickly. But the success of all these applications and apps depends on the quality of application development. Mobile apps have done a very vital role in all fields of life.

B. The Life Cycle of Developing Mobile Applications

For the development of mobile applications and building of mobile applications, this is the process in which all testing of the applications should be done according to the IDE environment. Quality assurance is done on thousands of mobiles applications to make sure all the applications are running properly. In this document of developing the applications, we are going to take through of introductory inspection process of building the mobiles applications [12].

C. Process Quality Assurance of Mobile Applications

The process of development of any software includes applications called software development life cycle [13]. In this process, all phases of DLC will be discussed for the development of mobile apps including development, design, inspiration, stabilization, deployment and maintenance of mobile applications if errors are bugs happen in the application [14].

TABLE 1
DIFFERENCES BETWEEN MANUAL AND AUTOMATED SOFTWARE TESTING

Manual Testing	Automated Testing
1. Time Consuming	Time Efficient
2. More human effort is required.	One-time human effort for creating the test scripts is enough.
3. Not accurate, due to room for human errors	More accuracy as less space for human error
4. Test cases cannot be reused	Supports reusability of test cases
5. More effective for functional testing and exploratory testing	Effective for regression testing, load testing & performance testing
6. Reduced short term cost (no automated testing tool is required) while increasing long term cost (maintenance).	Increased short term cost (automated testing tool) while reducing long term cost (maintenance).

D. Considerations Quality Assurance of Mobile Applications

There are lots of considerations when applications are developed especially the traditional web applications and desktop applications are developed [15].

1) *Mobile development apps*: There is no difference between developing mobile application life cycle in compar-

ison with web or desktop application [15]. There are five phases as we do in the development of web and desktop applications [16].

E. Inception Quality Assurance of Mobile Applications

All mobile apps are started from an idea. This idea leads to the development of the application.

TABLE 2
SOFTWARE QUALITY FACTORS FOR COMPARATIVE ANALYSIS

Software Quality Factors	Description
Extensibility	The ability of software components to be added, modified and removed easily without badly affecting the existing system. Flexibility is its category focused on the ability of components to be added easily.
Maintainability	Maintainability is the ability to make changes for error corrections, supported by defined interfaces, documentations, code comments.
Performance	Performance is related to acceptable response time.
Scalability	Ability to respond in an acceptable time in increased load or stress.
Robustness	Robustness is the ability of software to keep working and remain available in failure states by backup plans, data and hardware.
Usability	Usability is the ability of the user to easily interact with the system using the user interface.
Platform compatibility	Software should run on several platforms like operating systems, browsers etc.
Testability	Testability refers to maximum and efficient code coverage by testing.
Correctness	Correctness is software should conform to requirements or specifications.

F. Design Quality Assurance of Mobile Applications

This is the second phase and usually defines the user experience such as the general layouts and design of the mobile apps. With the help of graphical design, the apps designs are tested [17].

G. Development Quality Assurance of Mobile Applications

This is the phase of mobile application development in which mobile applications are developed. This is the actual main phase of building the mobile applications.

H. Stabilization Quality Assurance of Mobile Applications

When the mobile application is developed then these apps are stabilized and tested and make it free from bugs and errors before deployment sessions. But mainly quality assurance starts from this phase.

When development is far enough along, QA usually begins to test the application and bugs are fixed. Often an application will go into a limited beta phase in which a wider user audience is given a chance to use it and provide feedback

and inform changes [18].

I. Deployment Quality Assurance of Mobile Applications

Once the application code is done tested and debug from various testing phases then it is deployed. There are different ways of deploying the project. A project can be deployed directly to the websites or designated apps on the real servers or these projects can be deployed on the file system on the local machines. Latter these projects can be uploaded onto the servers or executed on the desktop applications [19].

With the doubt or ubiquity level of inspection and interaction with mobile devices, everyone has an idea about the mobile app. Mobile applications have changed the criteria and way of communication not only with computing but with people all over the world [20].

The inception stage is about refining the idea for mobile development apps and mobile related applications. For creating successful apps, we need to understand some fundamental questions. Some things need to be considered before the app is launched in the public store [21].

J. Infrastructure Integration of Mobile Applications

The mainly the infrastructure which is related to the mobile apps is considered and which type of app is properly can be developed [22].

K. Value of Mobile Applications

How valuable this app is and how this app can do better changes in the life of users and how the users make it benef-

icent for all the users [23].

1) *Designing mobile applications*: Once the design and functionality of the app are determined and examined by the top-level software and hardware engineers, the next step is to start working on the user's experience of previous mobiles apps and what they want new in new mobile apps [24].

TABLE 3
AUTOMATED MOBILE APPLICATIONS TESTING TOOL

Software Quality Factors	Description	
Testing Tool	Testing Type	Platform
Dynodroid	Event driven testing	Android
Evodroid	System testing	Android
FSM Droid	GUI testing	Android
MobiGUITAR	GUI testing	Android
Renorax	Compatibility testing	C#, Python, VB.net
Reran	GUI, system, stress, and security testing	Android
Robotium	GUI, system, functional, and acceptance testing	Android
Appium	GUI and functional testing	Android, IOS
MonkeyTalk	Compatibility and functional testing	Android, IOS
UI Automator	Functional and GUI testing	Android

2) *UX design or user experience design*: These are designs simply done via mockups, wireframes by using different tools like Mocking Bird and Visio. UX designs can be created free from tension and without any problem with these tools and worry about the actual UI designs [25]. When UX mockups are created, it is very important to keep an eye on other platforms like apple, android, and windows phones [26].

3) *Apple*: These are the human interface guidelines.

4) *Android*: These guidelines are for the designs.

5) *Windows*: These are library designs for windows and users of windows mobile [27]. Each app in mobiles applications has a metaphor for switching between sections of the apps and applications. Different operating systems use different types of graphics. iOS uses a tab bar at the bottom of each application and android is used on the top of the applications but windows mobiles use panorama view.

Most of the hardware automatically dictates UX decisions and work according to the environment. For instance, iOS has never a physical button on the back but the company introduced the navigation controller metaphor [28].

More factors influence UX decisions. If the table displaying has a display of real estate, it can display more information. So, the apps should work on different screens and compress

into one for the tablet view [29].

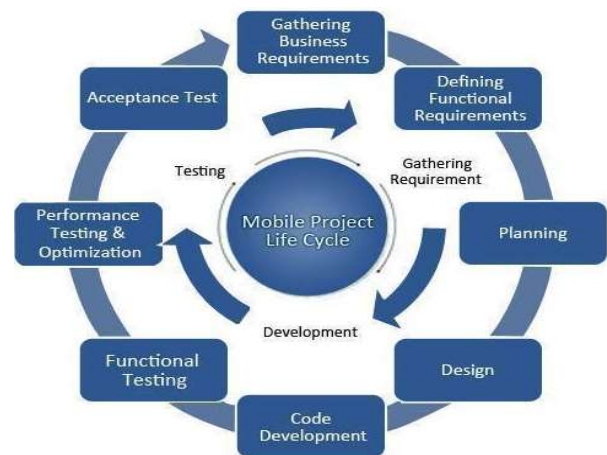


Fig. 1. Mobile software testing project life cycle

6) *User Interface (UI) design app*: According to fig 1 and the UX is designed and determined, the next step is to develop the UI design for the app and user.

UX is giving more black and white mockups. But in UI interface is where colours and graphics are finalized and introduced. If a good UI is created, it is very important to develop general and popular apps. But it is made sure, whenever UI and UX are created, they will use their platform and language [30]. If mobile application and app are well de-

signed, may still look interactive and different on different platforms.

7) *Development of mobile app*: The development phase of mobile applications starts in the early stages. If any idea has a maturity level on conception and inspiration then it is very easy to design and develop the mobile app and this idea validates the functionality and the assumptions about the understanding of the work [31].

8) *Prototype phase*: In this phase, the app is still in the conceptual phase and can be functional. There are major bugs that are still present but most of the parts of the mobile applications is working [32].

9) *Alpha phase*: Complete code and core functionality are tested in the alpha phase. But this is not fully testable or in this phase full requirements of the application can't be tested. But in this phase still, a lot of bugs are present and may create problems in functionality [33].

10) *Beta phase*: Most of the functionality is done and compete in this phase but a little bit of testing and bug fixing is left. But major issues may be still present in this phase [34].

11) *Release candidate phase*: All functionality is of a mobile app is tested and complete. But here in this phase, new bugs are released to the wild phase.

12) *Apple app store apps of iPhone*: Apple apps are available on the apple store and these apps can be downloaded by the iTunes store. It is a very excellent method to upload mobiles apps on the store and can be downloaded by the customers. It also allows the developers to make good market and distribution of their apps [35].

13) *House deployment of mobile apps and applications*: Basically, in house deployment is meant for the internal distribution of corporate apps and mobiles applications that are not available for the users by the app store.

14) *Android applications QA*: What are applications and apps which are belonged to the android store are signed before placing for distribution for the general users? Developers always signed their applications with certificates and with a private key. So, these certificates are providing a chain of authenticity. So, this chain bound the developer to release the application on a standard basis [36]. It is necessary that while developing the android application the developer should have to sign the certificate. The main purpose of the certificate is to make difference between the developer of the application and among the other developers of the application [5]. Android uses this information of permission between applications and all other components of the running android operating system.

In this fast-growing world, mobiles, mobile applications, smartphones and other communicational devices related to

mobile phones become most popular and usage of mobiles are increased in this new century. There are different operating systems are introduced in mobiles phone technology like ios, android, windows, blackberry, and Symbian. But people like most smartphones have android systems and apple I Phone. But there are a lot of challenges for the manufactures of hardware and software development for the mobiles to stay in completion otherwise they will be out from the race of technology [37]. There is also competition among the application developers of mobiles and mobile apps. So, the companies want long term success in the field of mobile phone technology. Mobile phones hardware and software along with different applications which are built-in mobile phones are all tested over several different platforms before launching to the global platform. Functional testing of the mobile's phones is done along with no function testing like security testing and usability testing. The basic purpose of mobile phone testing is to improve the mobile phone hardware, software, and applications along with apps. In this research project, we are going to test the mobile applications along with their competence in mobile testing technology. We can also learn mobile automation which is the main challenge nowadays. Many mobile phone applications testers are confused in understanding the hybrid mobile apps of the native companies. In the market different types of automation tools are available but different tools have their advantage. So that's very easy for the developers to select the best tool for the applications development. The quality of mobile applications for business use and especially for the company stakeholders can be highly trusted and if the application failure has happened it can take lead to serious failure and consequences. Mobiles applications become popular among people and these applications [38].

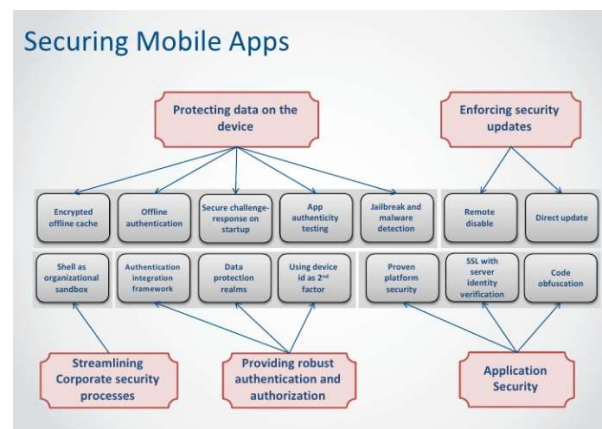


Fig. 2. The design structure of securing the mobile app from threats

L. Mobile Development Considerations of Apps

There is the main difference in developing the mobile application or app from the traditional desktop application or web application in terms of the architecture of the system and considerations should be aware of these given below.

M. Common Considerations of Mobile Apps

1) *Multitasking apps on the device:* The main purpose of mobile apps is to run with other applications like on the computer system all applications can be run at the same time. But here is the main challenge with the mobile application is to run apps with the other apps as well, so that this will become multitasking [39].

What are applications and apps which are belonged to the android store are signed before placing for distribution for the general users? [40]. So, this chain bound the developer to release the application on a standard basis. It is necessary that while developing the android application the developer should have to sign the certificate. The main purpose of the certificate is to make difference between the developer of the application and among the other developers of the application. Android uses this information of permission between applications and all other components of the running android operating system [41].

There are two categories of mobile devices, phones and tablets. To develop form factors for these applications are the same but application design format and screen sizes in development are different. The phone has a limited screen size and whereas the tablet has a greater and bigger screen size. So according to the sizes which vary from device to device, UI controls are developed according to the devices.

N. Device and OS Fragmentation of Mobile Apps

1) *Conceptualization and planning for mobile apps quality assurance:* There are different models which are introduced in the market. But all these models are different and from different companies. All these mobiles have different features. Due to difference, these mobiles can run some applications properly are sometimes this is not possible to run the mobile apps properly. Sometimes not all mobile have cameras as well. Some devices can play video and some cannot play video [42].

Quality assurance of mobile apps deals with high-quality apps is produced based on standards and deliver high-quality products line for the consumers. The mobile apps which are not according to standards and high quality are immediately removed from the mobiles and uninstall. Nowadays banks and all other businesses are run on a lie and all types of work of business are done on mobile phones

and customized apps. Users can use several different types of apps which are helpful in businesses to get immediate responses from these applications and make users records from these applications. Especially, the bank's application is designed according to customers need and customers can do a lot of transactions by using these apps and they do not need to go to the banks. With the help of these applications, they can sort out their problems quickly. But the success of all these applications and apps depends on the quality of application development. Mobile apps have done a very vital role in all fields of life [43].

For the development of mobile applications and building of mobile applications, this is the process in which all testing of the applications should be done according to the IDE environment. Quality assurance is done on thousands of mobiles applications to make sure all the applications are running properly. In this document of developing the applications, we are going to take through of introductory inspection process of building the mobiles applications [44].

O. Considerations

There are a lot of considerations when applications are developed especially the traditional web applications and desktop applications are developed.

P. Mobile Development Apps

There is no difference between developing mobile application life cycle in comparison with web or desktop application. There are five phases as we do in the development of web and desktop applications.

Q. Inception

All mobile apps are started from an idea. This idea leads to the development of the application [45].

R. Design and Quality Assurance

When applications are designed, these designs got the user attention because of different screen ratios and sizes while designing an application with UI. So, these screen sizes can be considered differently.

S. Development Apps and Quality Assurance

When mobile apps features are tested, these features can be tested with the help of codes. For instance, if the mobile app features are tested, the camera is tested first because of different features [46]. When features are tested, it is made sure to get the proper support of the operating system.

T. Device-specific Resources and Quality Assurance

The main purpose of mobile apps is to run with other applications like on the computer system all applications can be run at the same time. But here is the main challenge with the mobile application is to run apps with the other apps as well, so that this will become multitasking [47].

So, in mobile applications only one app can be run in the foreground and the rest of the other apps will be opened but stay behind, in this way battery consumption can be saved as well. All different platforms handle multitasking apps differently.

The main purpose of mobile apps is to run with other applications like on the computer system all applications can be run at the same time. But here is the main challenge with the mobile application is to run apps with the other apps as well, so that this will become multitasking [48].

So, in mobile applications only one app can be run in the foreground and the rest of the other apps will be opened but stay behind, in this way battery consumption can be saved as well. All different platforms handle multitasking apps differently.

U. SQLite Database for Mobile Apps

Both iOS and Android systems use the SQLite database for storing data from different applications.

But windows phones are not using a database but versions that are launched in the market are using a database. Windows Phone 7.1 and 8 use a local database engine to support LINQ to SQL but it does not support SQL queries.

So, in mobile applications only one app can be run in the foreground and the rest of the other apps will be opened but stay behind, in this way battery consumption can be saved as well. All different platforms handle multitasking apps differently.

V. Form Factor for Mobile Apps

There are two categories of mobile devices, phones and tablets. To develop form factors for these applications are

the same but application design format and screen sizes in development are different. The phone has a limited screen size and whereas the tablet has a greater and bigger screen size. So according to the sizes which vary from device to device, UI controls are developed according to the devices [49].

W. Device and OS Fragmentation of Mobile Apps

It is very important to take different devices throughout the whole software development and these apps will show the proper behaviour of devices.

X. Conceptualization and Planning for Mobile Apps Quality Assurance

There are different models which are introduced in the market. But all these models are different and from different companies. All these mobiles have different features. Due to difference, these mobiles can run some applications properly are sometimes this is not possible to run the mobile apps properly. Sometimes not all mobile have cameras as well. Some devices can play video and some cannot play video.

Y. Design and Quality Assurance

When applications are designed, these designs got the user attention because of different screen ratios and sizes while designing an application with UI. So, these screen sizes can be considered differently.

1) *Development Apps and Quality Assurance* When mobile apps features are tested, these features can be tested with the help of codes. For instance, if the mobile app features are tested, the camera is tested first because of different features. When features are tested, it is made sure to get the proper support of the operating system [50].

2) *Testing Quality Assurance of Mobile Applications* For testing purposes, the applications are tested on the different platforms and then actually tested on the actual mobiles and different devices so that testing and quality assurance may be done properly for all these applications.

TABLE 4
QUALITY FACTORS ACHIEVED BY AUTOMATED TESTING OF MOBILE APPS

Software Quality Assurance Factors								
Extensibility	Maintainability	Performance	Scalability	Robustness	Usability	Platform Compatibility	Testability	Correctness
Dynodroid	Yes		Yes		Yes		Yes	
Evodroid		Yes	Yes	Yes		Yes	Yes	
FSM Droid		Yes		Yes		Yes		Yes
MobiGUITAR			Yes		Yes		Yes	
Renorax			Yes		Yes		Yes	
Reran		Yes	Yes		Yes	Yes		Yes
Robotium			Yes		Yes		Yes	
Appium				Yes			Yes	
MonkeyTalk				Yes			Yes	
UI Automator				Yes			Yes	

3) IOS Considerations of Mobile Apps

4) *Device-specific resources and quality assurance:* The main purpose of mobile apps is to run with other applications like on the computer system all applications can be run at the same time. But here is the main challenge with the mobile application is to run apps with the other apps as well, so that this will become multitasking [51].

So in mobile applications only one app can be run in the foreground and the rest of the other apps will be opened but stay behind, in this way battery consumption can be saved as well. All different platforms handle multitasking apps in a different ways.

5) *Android considerations:* The main purpose of mobile apps is to run with other applications like on the computer system all applications can be run at the same time. But here is the main challenge with the mobile application is to run apps with the other apps as well, so that this will become multitasking.

So in mobile applications only one app can be run in the foreground and the rest of the other apps will be opened but stay behind, in this way battery consumption can be saved as well. All different platforms handle multitasking apps differently [52].

III. FINDINGS AND DISCUSSIONS

According to Table 4, Evodroid and Renorax aid to achieve quality factors “extensibility” and “maintainability”. “Performance” of the AUT can be enhanced by using three tools i.e., FSM Droid, Reran, and Robotium. Among all the tools, only Reran aims to achieve the “scalability” of the AUT. “Robustness” can be achieved by five tools i.e., Dynodroid, Evodroid, FSM Droid, MobiGUITAR and Reran. Many of the tools assure “usability” of the AUT i.e., Dynodroid, FSM Droid, Reran, Robotium, Appium, and UI Automator. The “platform compatibility” testing is supported by only two tools, namely, Renorax and MonkeyTalk. “Testability” of the app

can be verified and enhanced using four tools, namely, Dynodroid, Evodroid, FSM Droid, and MobiGUITAR. Most of the tools, namely, Dynodroid, Evodroid, MobiGUITAR, Reran, Robotium, Appium, MonkeyTalk, and UIAutomator assure the “correctness” of the AUT.

Fig 1 presents a graph showing the results of this comparative study. Ten dominant automated testing tools for mobile applications are considered for this study. Each tool focused on one or more quality factors to achieve or enhance the quality of apps under test. Moreover, Fig 3 shows an overall trend of quality factors achieved by using automated testing. The quality factor of “correctness” will be achieved using almost every automated testing tool. “Usability” is also a major aspect of mobile apps which can be evaluated and achieved by using approximately 60% of the available software testing tools. Approximately 50% of these tools focus on achieving desired or optimum level of “robustness” of mobile apps. A close to the average percentage of testing tools attain quality factors of “testability” and “performance” in the app under test.



Fig. 3. Frequency of software quality factors achieved using automated testing tools for mobile applications

A. Windows Phone Considerations for Mobile Apps

1) *SQLITE database for mobile apps*: Both iOS and Android systems use the SQLITE database for storing data from different applications.

But windows phones are not using a database but versions that are launched in the market are using a database. Windows Phone 7.1 and 8 use a local database engine to support LINQ to SQL but it does not support SQL queries [53].

B. Analysis of Mobile Application Quality Assurance

In a complex and fragmented environment, mobile application products help you to improve the quality assurance of different mobile applications. Mobile quality assurance delivers feedback from users at a high-quality level. This product also helps in validating and tracking the user experience how the user can like and appreciate the product. There are lots of goals that can be achieved for the quality assurance of mobile applications. Improve the quality of mobile apps with user experience and make changes in next coming applications. This user experience can improve the quality assurance of mobiles applications by getting bugs, crash reports, users' feedback and user experience. Try to reduce risks which can be assessing issues and reducing time [54]. Make the report of the users experience properly, make the feedback report and collect other information relevant to the mobile application and then develop a strategy to minimize all these problems.

How to improve the application is a very important role in mobile application quality assurance. A technical developer and owner can use different sources like feedback and crash reports. Developers are very helpful in building great mobile applications. A mobile application developer can easily get feedback from its existing application. This role goes to the application developer which uncovers challenges and user experience. When mobile users are using mobile applications can send different reports to the developers and the developers easily handle the problems created in the applications. In this research project, we are going to test the mobile applications along with their competence in mobile testing technology [55]. We can also learn mobile automation which is the main challenge nowadays. Many mobile phone applications testers are confused in understanding the hybrid mobile apps of the native companies. In the market different types of automation tools are available but different tools have their advantage. So that's very easy for the developers to select the best tool for the applications development [56].

IV. CONCLUSION AND FUTURE WORK

In this publication of the research paper, we are going to represent mobile application quality assurance, tools, platforms and launch the mobile application after testing by the different devices. There are five types of testing is done like functional testing, performance and load testing by which we can do acceptance of testing. For each category, we need to find out a set of tools and display the distribution on a commercial level and make it onto open source for everyone. There are more than 80 tools that can be used to develop the mobile application. The quality of mobile applications for business use and especially for the company stakeholders can be highly trusted and if the application failure happens it can take to lead serious failure and consequences. Mobiles applications become popular among people and these applications in games, medicine, science and on different topics are launched. But the applications which are relevant to business are very important and attractive to stakeholders as well. For other applications and apps of mobiles, it should be emphasized the users to be easily used by them. The basic goal of this application is to identify and recognize the approaches that address the issues of quality assurance of mobile applications and apps. The basic purpose of mobile phone testing is to improve the mobile phone hardware, software, and applications along with apps. In this research project, we are going to test the mobile applications along with their competence in mobile testing technology. Several tools are proposed and implemented for the testing of mobile apps. In this research study, these tools are evaluated focusing on identifying the quality factors they aid to achieve in the apps under test. Moreover, overall trends of essential quality factors achieved using automated testing tools are measured. This study revealed that automated testing provides the best support for assurance of usability, correctness and robustness. An average number of tools aid to assure testability and performance. However, for assurance of extensibility, maintainability, scalability, and platform compatibility, only a few tools are available. In automated testing of mobile applications, further research can be done to propose an automated mobile apps testing tool that aims to achieve all quality factors mentioned in Table 2. A similar analysis can be made by considering testing tools for other mobile operating systems too, e.g., windows. A comparative analysis can also be done on the quality of apps of different mobile operating systems based on the automated testing tools of each platform. Moreover, based on the tools identified from this study, revised and enhanced solutions can be proposed for achieving the maximum number of quality attributes in the AUT.

REFERENCES

- [1] P. Rathi and V. Mehra, "Analysis of automation and manual testing using software testing tool," *International Journal of Innovations & Advancement in Computer Science*, vol. 4, pp. 709-713, 2015.
- [2] D. Amalfitano, A. R. Fasolino, P. Tramontana, B. D. Ta, and A. M. Memon, "MobiGUITAR: Automated model-based testing of mobile apps," *IEEE Software*, vol. 32, no. 5, pp. 53-59, 2014. doi: <https://doi.org/10.1109/ms.2014.55>
- [3] X. Wang and G. He, "The research of data-driven testing based on QTP," in *9th International Conference on Computer Science & Education*, Vancouver, BC, 2014. doi: <https://doi.org/10.1109/ICCSE.2014.6926625>
- [4] K. M. Mustafa, R. E. Al-Qutaish, and M. I. Muhairat, "Classification of software testing tools based on the software testing methods," in *Second International Conference on Computer and Electrical Engineering*, Dubai, UAE, 2009. doi: <https://doi.org/10.1109/ICCEE.2009.9>
- [5] H. Kaur and G. Gupta, "Comparative study of automated testing tools: Selenium, quick test professional and testcomplete," *International Journal of Engineering Research and Applications*, vol. 3, no. 5, pp. 1739-1743, 2013.
- [6] G. Saini and K. Rai, "Software testing techniques for test cases generation," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 9, pp. 261-265, 2013.
- [7] N. Islam. (2016) A comparative study of automated software testing tools. [Online]. Available: <https://bit.ly/3E197Y0>
- [8] K. Shikha and K. Bahl, "Software testing tools & techniques for web applications," *International Journal of Engineering and Technical Research (IJETR)*, vol. 3, no. 5, pp. 315-318, 2015.
- [9] S. Sharma and M. Vishawjyoti, "Study and analysis of automation testing techniques," *Journal of Global Research in Computer Science*, vol. 3, no. 12, pp. 36-43, 2012.
- [10] A. Jain, M. Jain, and S. Dhankar, "A comparison of RANOREX and QTP automated testing tools and their impact on software testing," *IJEMS*, vol. 1, no. 1, pp. 8-12, 2014.
- [11] T. Xie, "Improving effectiveness of automated software testing in the absence of specifications," in *22nd International Conference on Software Maintenance*, Philadelphia, PA, 2006. doi: <https://doi.org/10.1109/ICSM.2006.31>
- [12] S. Gunasekaran and V. Bargavi, "Survey on automation testing tools for mobile applications," *International Journal of Advanced Engineering Research and Science*, vol. 2, no. 11, pp. 2349-6495, 2015.
- [13] L. Gomez, I. Neamtui, T. Azim, and T. Millstein, "Reran: Timing-and touch-sensitive record and replay for android," in *35th International Conference on Software Engineering (ICSE)*, San Francisco, CA, 2013. doi: <https://doi.org/10.1109/ICSE.2013.6606553>
- [14] A. Machiry, R. Tahiliani, and M. Naik, "Dynodroid: An input generation system for android apps," in *Proceedings of the 9th Joint Meeting on Foundations of Software Engineering*, New York, NY, 2013. doi: <https://doi.org/10.1145/2491411.2491450>
- [15] T. Su, "FSMdroid: Guided GUI testing of android apps," in *ACM 38th International Conference on Software Engineering Companion (ICSE-C)*, Austin, TX, 2016.
- [16] N. J. Wahl, "An overview of regression testing," *ACM SIGSOFT Software Engineering Notes* New York, NY, vol. 24, no. 1, pp. 69-73, 1999. doi: <https://doi.org/10.1145/308769.308790>
- [17] Y. Zhauniarovich, A. Philippov, O. Gadyatskaya, B. Crispo, and F. Massacci, "Towards black box testing of android apps," in *10th International Conference on Availability, Reliability and Security*, Toulouse, France, 2015. doi: <https://doi.org/10.1109/ARES.2015.70>
- [18] S. Zein, N. Salleh, and J. Grundy, "A systematic mapping study of mobile application testing techniques," *Journal of Systems and Software*, vol. 117, pp. 334-356, 2016. doi: <https://doi.org/10.1016/j.jss.2016.03.065>
- [19] Q. Do, G. Yang, M. Che, D. Hui, and J. Ridgeway, "Regression test selection for android applications," in *Proceedings of the International Conference on Mobile Software Engineering and Systems*, New York, NY, 2016. doi: <https://doi.org/10.1145/2897073.2897127>
- [20] C. Bernaschina, R. Fedorov, D. Frajberg, and P. Fraternali, "A framework for regression testing of outdoor mobile applications," in *4th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, Buenos Aires, Argentina, 2017. doi: <https://doi.org/10.1109/MOBILESoft.2017.13>
- [21] D. Kumar and S. Saxena, "Android application memory leakage detection approach," *The Engineering Journal of Application & Scopes*, vol. 2, no. 1, pp. 59-69, 2017.
- [22] N. u. Saqib and S. Shahzad, "Functionality, performance, and compatibility testing: A model based approach," in *Inter-*

- national Conference on Frontiers of Information Technology (FIT)*, Islamabad, Pakistan, 2018. doi: <https://doi.org/10.1109/FIT.2018.00037>
- [23] W. Choi, K. Sen, G. Necul, and W. Wang, "Detreduce: Minimizing android gui test suites for regression testing," in *40th International Conference on Software Engineering (ICSE)*, Gothenburg, Sweden, 2018. doi: <https://doi.org/10.1145/3180155.3180173>
- [24] K. S. Arif and U. Ali, "Mobile application testing tools and their challenges: A comparative study," in *2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, Sukkur, Pakistan, 2019. doi: <https://doi.org/10.1109/icomet.2019.8673505>
- [25] L. Deng, J. Offutt, and D. Samudio, "Is mutation analysis effective at testing android apps?" in *IEEE International Conference on Software Quality, Reliability and Security (QRS)*, Prague, Czech Republic, 2017. doi: <https://doi.org/10.1109/QRS.2017.19>
- [26] L. Deng, N. Mirzaei, P. Ammann, and J. Offutt, "Towards mutation analysis of android apps," in *Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, Graz, Austria, 2015. doi: <https://doi.org/10.1109/ICSTW.2015.7107450>
- [27] M. Linares-Vásquez, G. Bavota, M. Tufano, K. Moran, M. Di Penta, C. Vendome, C. Bernal-Cárdenas, and D. Poshyanyk, "Enabling mutation testing for android apps," in *Proceedings of the 11th Joint Meeting on Foundations of Software Engineering*, Paderborn, Germany, 2017. doi: <https://doi.org/10.1145/3106237.3106275>
- [28] Z. Zhou, C. Sun, J. Lu, and F. Lv, "Research and implementation of mobile application security detection combining static and dynamic," in *10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, Changsha, China, 2018. doi: <https://doi.org/10.1109/icmtma.2018.00065>
- [29] X. Huang, A. Zhou, P. Jia, L. Liu, and L. Liu, "Fuzzing the android applications with HTTP/HTTPS network data," *IEEE Access*, vol. 7, pp. 59951-59962, 2019. doi: <https://doi.org/10.1109/access.2019.2915339>
- [30] S. Yang, D. Yan, and A. Rountev, "Testing for poor responsiveness in android applications," in *1st International Workshop on the Engineering of Mobile-enabled Systems (MOBS)*, San Francisco, CA, 2013. doi: <https://doi.org/10.1109/mobs.2013.6614215>
- [31] H. Muccini, A. Di Francesco, and P. Esposito, "Software testing of mobile applications: Challenges and future research directions," in *7th International Workshop on Automation of Software Test (AST)*, Zurich, Switzerland, 2012. doi: <https://doi.org/10.1109/IWAST.2012.6228987>
- [32] R. Jabbarvand, J.-W. Lin, and S. Malek, "Search-based energy testing of android," in *IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, Montreal, QC. IEEE, 2019, pp. 1119-1130.
- [33] Y. Wang and Y. Alshboul, "Mobile security testing approaches and challenges," in *First Conference on Mobile and Secure Services (MOBISSECSERV)*, Gainesville, FL, 2015. doi: <https://doi.org/10.1109/MOBISSECSERV.2015.7072880>
- [34] Z. Liu, "Research of performance test technology for big data applications," in *International Conference on Information and Automation (ICIA)*, Hailar, China, 2014. doi: <https://doi.org/10.1109/ICInfA.2014.6932625>
- [35] J. Huang, L. Niu, J. Zhan, X. Peng, J. Bai, and S. Cheng, "Technical aspects and case study of big data based condition monitoring of power apparatuses," in *PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, Hong Kong, China, 2014. doi: <https://doi.org/10.1109/APPEEC.2014.7066164>
- [36] N. Li, A. Escalona, Y. Guo, and J. Offutt, "A scalable big data test framework," in *8th International Conference on Software Testing, Verification and Validation (ICST)*, Graz, Austria, 2015. doi: <https://doi.org/10.1109/ICST.2015.7102619>
- [37] M. Scavuzzo, D. A. Tamburri, and E. Di Nitto, "Providing big data applications with fault-tolerant data migration across heterogeneous NoSQL databases," in *ACM 2nd International Workshop on Big Data Software Engineering (BIGDSE)*, Austin, TX, 2016. doi: <https://doi.org/10.1109/BIGDSE.2016.013>
- [38] C. Xu, M. Holzemer, M. Kaul, and V. Markl, "Efficient fault-tolerance for iterative graph processing on distributed dataflow systems," in *32nd International Conference on Data Engineering (ICDE)*, Helsinki, Finland, 2016. doi: <https://doi.org/10.1109/ICDE.2016.7498275>
- [39] J. Ding, X.-H. Hu, and V. Gudivada, "A machine learning based framework for verification and validation of massive scale image data," *IEEE Transactions on Big Data*, vol. 7, no. 2, pp. 451-467, 2017. doi: <https://doi.org/10.1109/TBDATA.2017.2680460>
- [40] L. Lundberg, H. Grahn, D. Ilie, and C. Melander, "Cache support in a high performance fault-tolerant distributed storage

- system for cloud and big data," in *International Parallel and Distributed Processing Symposium Workshop*, Hyderabad, India, 2015. doi: <https://doi.org/10.1109/IPDPSW.2015.65>
- [41] E. M. Fredericks and R. H. Hariri, "Extending search-based software testing techniques to big data applications," in *ACM 9th International Workshop on Search-Based Software Testing (SBST)*, Austin, TX, 2016. doi: <https://doi.org/10.1109/SBST.2016.016>
- [42] J. Lin, F. Liang, X. Lu, L. Zha, and Z. Xu, "Modeling and designing fault-tolerance mechanisms for mpi-based mapreduce data computing framework," in *First International Conference on Big Data Computing Service and Applications*, Redwood City, CA, 2015. doi: <https://doi.org/10.1109/BigDataService.2015.33>
- [43] J. Morán, A. Bertolino, C. de la Riva, and J. Tuya, "Automatic testing of design faults in mapreduce applications," *IEEE Transactions on Reliability*, vol. 67, no. 3, pp. 717-732, 2018. doi: <https://doi.org/10.1109/TR.2018.2802047>
- [44] X. Zhao and X. Gao, "An ai software test method based on scene deductive approach," in *International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, Lisbon, Portugal, 2018. doi: <https://doi.org/10.1109/QRS-C.2018.00017>
- [45] C. Jiang, S. Huang, and Z.-w. Hui, "Metamorphic testing of image region growth programs in image processing applications," in *International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, Lisbon, Portugal, 2018. doi: <https://doi.org/10.1109/QRS-C.2018.00026>
- [46] M. Shi and R. Yuan, "Mad: A monitor system for big data applications," in *International Conference on Intelligent Science and Big Data Engineering*, Suzhou, China, 2015. doi: https://doi.org/10.1007/978-3-319-23862-3_30
- [47] Y. Yang, J. Ai, and F. Wang, "Defect prediction based on the characteristics of multilayer structure of software network," in *International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, Lisbon, Portugal, 2018. doi: <https://doi.org/10.1109/QRS-C.2018.00019>
- [48] R. Malhotra, "Comparative analysis of statistical and machine learning methods for predicting faulty modules," *Applied Soft Computing*, vol. 21, pp. 286-297, 2014. doi: <https://doi.org/10.1016/j.asoc.2014.03.032>
- [49] G. Akash, O. T. Lee, S. M. Kumar, P. Chandran, and A. Cuzzocrea, "Rapid: A fast data update protocol in erasure coded storage systems for big data," in *17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CC-GRID)*, Madrid, Spain, 2017. doi: <https://doi.org/10.1109/CCGRID.2017.115>
- [50] G. Shi and H. Wang, "Research on big data real-time public opinion monitoring under the double cloud architecture," in *Second International Conference on Multimedia Big Data (BigMM)*, Taipei, Taiwan, 2016. doi: <https://doi.org/10.1109/BigMM.2016.35>
- [51] X. Liu, X. Fan, and J. Li, "A novel parallel architecture with fault-tolerance for joining bi-directional data streams in cloud," in *International Conference on Cloud Computing and Big Data*, Fuzhou, China, 2013.
- [52] G. Iuhasz and I. Dragan, "An overview of monitoring tools for big data and cloud applications," in *International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, Timisoara, Romania, 2015. doi: <https://doi.org/10.1109/SYNASC.2015.62>
- [53] S. Ganguly, A. Consul, A. Khan, B. Bussone, J. Richards, and A. Miguel, "A practical approach to hard disk failure prediction in cloud platforms: Big data model for failure management in datacenters," in *Second International Conference on Big Data Computing Service and Applications (BigDataService)*, Oxford, UK, 2016. doi: <https://doi.org/10.1109/BigDataService.2016.10>
- [54] Y. Wang, Y. Shen, H. Wang, J. Cao, and X. Jiang, "MTMR: Ensuring mapreduce computation integrity with merkle tree-based verifications," *IEEE Transactions on Big Data*, vol. 4, no. 3, pp. 418-431, 2016. doi: <https://doi.org/10.1109/TBDDATA.2016.2599928>
- [55] D. Puthal, S. Nepal, R. Ranjan, and J. Chen, "Dlself: A dynamic key-length-based efficient real-time security verification model for big data stream," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 2, pp. 1-24, 2016. doi: <https://doi.org/10.1145/2937755>
- [56] L. E. B. Villalpando, A. April, and A. Abran, "Performance analysis model for big data applications in cloud computing," *Journal of Cloud Computing*, vol. 3, no. 1, pp. 1-20, 2014. doi: <https://doi.org/10.1186/s13677-014-0019-z>