 doi:10.20944/preprints201912.0060.v1

Focused Software Quality Assurance in Mobile Applications

Mehreen Sirshar
Faculty of Software Engineering
Fatima Jinnah Women University
Rawalpindi, Pakistan
mehreensirshar@fjwu.edu.pk

Sania Imran
Software Engineering
Fatima Jinnah Women University
Rawalpindi, Pakistan
saniaimran3675@gmail.com

Faiqa Mehboob
Software Engineering
Fatima Jinnah Women University
Rawalpindi, Pakistan
faiqamehboob6191@gmail.com

Abstract—As the time passes the modification in technology world lead to the evaluation in mobile application as well. With evaluation in mobile industry it is an open challenge for software quality researcher that how to enhance software quality to meet the needs of changes? Quality assurance play a key role in differentiating good application from bed application. With the continuous evaluation of mobile application developing process should be quick and efficient to comply with user requirements and satisfaction. While the listed requirement leads to bad design choices known as anti-patterns, which in turn affect the reliability of the code. A tool based method PAPRIKA is used in the proposed re-search to identify and monitor these anti-patterns together with a two-step assessment model for software quality assurance and object oriented software quality matrix.

Index Terms—mobile app, software quality anti-patterns,

I. Introduction

Software evolve with the changing technologies to fulfill advanced requirement, to enhance software design and to fit to changing environment. With the gradual aging of software, the software quality gets worse regardless of what kind of changed occur [1]. Along with software evaluation the quality of software depreciates due to introduction of deprived software design and implementation selections. Researchers are studying this factors and have proposed number of techniques and tools to detect these anti-patterns [2]. To find out the quality of mobile application the object oriented anti-patterns are under studied factor. Focus of discovery is to identify from object oriented principle like functional breakdown. Anti-patterns and code smells might transpire regardless of the software development kit accessibility for mobile software application because of restrictions on re-sources like CPU or screen sizes, memory. With the evaluation in mobile application due to presence of anti-patterns and Object Oriented code smells for example Blob class, the quality of software decreases at the constant level. Deviation of software matrices with the passage of time is the main technique used concerning software evaluation with respect to software quality. In proposes work to observe the evaluation of mobile application while approaching mobile software quality by considering anti-patterns, a tool based automated technique is used. In proposed technique the metric based detection approach is introduced rather than unit evaluation of mobile application. In PAPRIKA quality tracking and automatic anti-patterns detection are automatically included. The main point in arrears this approach is to find out how variation in software artifact size and anti-pattern effect the quality of software. Study showed that a big data of anti-patterns is hosted at formation of software artifacts while some of them hosted while changing being performed on software artifacts while evaluation. Demand of mobile application is increasing rapidly while advance application is being developed. According to statistical analysis until 2020 the mobile application industry will grow from 70 billion US dollars in 2015 to 189 billion US dollars [3]. Due to increase in demand of mobile application the need for high quality assurance for these application is also the main concerns of developers and software quality assurance engineers. Focus on this problem a new two-way evaluation called two step evaluation approach is proposed known as FIT4.The mobile specific problem arises after the re-lease of mobile application are addressed and prevent by using this approach. Two-way evaluation: mobile-specific test and mobile specific inspection are used. The mobile specify inspection is responsible for preventing failure while mobile specific test is used for finding failure during development of mobile applications. Quality assurance help developer to have best quality track of his software application. Lack of software quality in mobile applications are due to variety of different platforms that are in mobile market, example of these platform include Symbian [4] Android [5], window mobile [6], IOS [7]. Design module and architecture for each of these platforms are unique and different. With the advancement in technology mobile market is evaluating and competition is increasing that's why mobile application development cycle are reserved diminutive. Which in turn response in huge market demand of mobile application with unique features and introduction of new approaches to control their quality by applying new methods for quality assurance. Software quality metric is introduced in this paper to measure the software quality of mobile applications at the testing stage. Quality metrics are used to define nonfunctional requirement during requirement analysis. To meet up the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 December 2019 doi:10.20944/preprints201912.0060.v1

metrics requirement mobile software application is tested thoroughly during test phase. The objective of using this approach is to estimate the current basis code metrics for the further future procedure in the mobile software application development.

II. Literature Review

Rapidly evolving hardware and software technology is increasing day by day. Everything related to technology is presented in the form of mobile applications. Quality assurance of mobile application is important to differentiate good apps from bad ones. To ensure the good quality of any mobile application there are numerous automated testing tools available that helps in defining and testing the quality of applications. This research [8] aims at experimenting with different automated testing tools for mobile applications on window platform. These automated tools are useful to measure the functional and non-functional requirements and to check the platform compatibility. This conclude that automated tools increase effectiveness and improve accuracy to provide good quality window based mobile applications. Nowadays, a smart mobile phones are used by almost every human being on earth. In-crease in dependability on mobile application increased the needs of testing and as-sure good quality products. To assure high quality, mobile specific failure classification patterns are used. This work [9] evaluates the classification in terms of completeness in fault and failure classes. The proposed methodology used to define classification patterns is FIT4Apps (Focus Inspection and Tests for Mobile Application) which ensure high quality mobile applications. This research concludes that how usable, how complete and how effective the classification is. The increasing demand for mobile applications and technology leads many companies in the production of software to demonstrate the good quality of their products. Assuring the good quality of mobile applications is highly related issue since the application failure. This paper [10] conducts the survey that focus on mobile applications testing and testing tools to analyze good quality. In terms of unit, integration, regression and process analysis, several problems are examined. Defects are the major contributors to bad software quality products. In proposed work [11] model named HIRER is presented to predict the defects in android file using functional and systematic data in mobile applications as defect detection is part of software quality assurance. Defect prediction can improve software application effectiveness and reduce cost. Projected model HIRER [11] calculate defects in source code spontaneously. HIRER uses four android applications from Google GIT and follow the same methodology in defect marking tool DCRS. It detects whether android file contains any defect by observing GIT change log. This research [12] describes an extensible architecture, i.e. interaction through heterogeneous architectures and theoretical prototypes, that displays and combines simultaneous cross-platform testing with performance measurements. It is possible to perform manual analysis, but this is time consuming, tedious and prone to error. It means that there is an automated testing method for increasing the manual labor involved in testing and therefore validating good quality applications. Many software quality assurance procedures are introduced to enhance good quality software applications. A software quality assurance technique is proposed in [13] by mechanizing fragments of a mobile cross-platform regression test. The methodology used to accomplish the planned goal is Design-science research paradigm. For experimental purpose secure environment with very basic mobile application under related test cases is used in order to concentrate on core challenges related to quality in cross-platform testing. Mobilette's architecture is proposed to support additional commands features as well as error handling in unpredictable situations. Mobilette also give advantage [13] that test can run in parallel on multiple devices on basis of performance test executed on Mobilette.

III. Proposed Methodology

A. PAPRIKA: A Tool based approach for detection of software anti-patterns

This approach is completed in four steps. In first phase APK file of respected mobile application is extracted in order to extract metadata which include application name, packages and also the application code. Mobile Application ranking, downloads numbers, this additional metadata also received from google play store and then passes to system as argument. This information is then automatically passed on-ward in order to figure out a code model which include classes with attributes and methods. In second step this model information is stowed to a graph database. Third step include detection of anti-patterns in the code while interrogating the respect-ed graph. Final step in this tool based approach is to calculate the software quality evaluation score on the basis of provided vision history.

Term	Units	Explanation
Name	All	Entity Name
Application Key	All	App exclusive id
Ranking	Application	Ranking on Play store
Download date	Application	APK date of down-load
App Size	Application	APK Size
App Type	Variables	Object type of variables
Return types	methods	Methods return type
Location	Arguments	Location of argument in method

Step 1: Metric collection from mobile application

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 December 2019 doi:10.20944/preprints201912.0060.v1

ARTIFACT

This phase includes creating mobile application model and extracting software quality metrics from the software artifact. The model in this phase is made on basis of analyzing the code byte while complemented with the properties composed from mobile play store. These are the entities on basis of which model is designed: Application, classes, Methods, Attributes, Variables, External Classes and External Methods. Table 1 define the properties that are attributed to these entities. Many mobile applications on google play are too complicated to size optimization and make the process slow down. Mostly attributes, methods and classes are retitled as single letters that's why just on lexical data we cannot rely in order to calculate software quality metrics so sidestep approaches concept is introduced.

STEP 2: GRAPH MODEL FROM PAPRIKA MODEL

In this phase quality graph model which is generated in phase one is introduced to database for further classification on data. Graph database named NEO4Jsss is used in this research work. Reason behind using this database this that on combination with CYPHER query language it gives good quality performance on heavy dataset

STEP 3: ANTIPATTERNS DETECTION FROM GRAPH MODEL

In this second last step of PAPRIKA approach the anti-patterns are detected from the graph database. Database query language CYPHER is used to detect software anti-patterns. Respected result is grouped in versions. The object oriented anti-patterns are detected by using a threshold value, which classify high value from set of other common values. Outlier are defined for the whole dataset in order to define thresholds value while using Tukey Box plot. Values lies upper whisker pointed as very high while values lie below whisker identify as the lower one. Threshold contain the representation of overall mobile application present in dataset not only the contemporary analyze application.

STEP 4: MOBILE SOFTWARE QUALITY CALCULATION FROM DISCOVERED ANTIPATTERNS

In final step discovered anti-patterns from different version of mobile application are used to compute mobile software quality. Linear regression is used to build a model in order to find the software quality score. The linear regression model is responsible for representing the relationship between the mobile application size and number of anti-patterns. Derogation of squared residuals is used in order to calculate the linear regression. The software quality score in PAPRIKA is computed by the additive inverse of the respected residual having the value for the size of mobile application size as well as the number of software anti-patterns. Negative value of residual value suggests good software quality because the observed version have small quantity of anti-patterns. On the other hand, presence of positive residual lead to poor software quality because in the observed studied version amount of anti-patterns are greater than the size norm (linear regression) of mobile application

B. Software Quality Metric (OO Metric)

Object oriented programming is mostly used in development of Mobile applications such as, Objective –C or Java etc. Object oriented metric gain its popularity since it is one of the best approach to assess quality assurance of mobile software applications. This method is widely used and have a very smooth possibility in further modification as well. This linearity lead Object Oriented metric to the further creation of tool based method example as DECOR or IPLASMA Object oriented matrix detect any defect in coding pin point it and use the emergency approach to remove that defect helping mobile application to work properly hence enhancing software quality assurance of mobile application. SAMOA tool is use to analyze the mobile app specification and allow software quality researcher as well to analyze their mo-bile application from the source code. The purpose of this approach to use is that using it can reduce the possibilities of any defect that can be present at coding level, thus removing that possibility of defect at very basic level. Source code of mobile software matrices are tested during tested phase under the supervision of software quality assurance researcher for their role part in development of mobile software. Based on the result the quality attributes of the evaluating mobile application are listed down.

C. A Two-Step Evaluation Approach (FIT4Apps)

Two step evaluation model for software quality assurance for mobile application work on the basis of pre-define set of values i.e. concepts. It basically work on problem before it arises actually in work environment. In two step model the hypothesis are created to support the future problem and its solution. Least possible example of this model working scenario is this that assume the software failure exist before it's released. To ensure the quality of soft-ware we work on hypothesis that how to ensure that low chance of software failure exist? How to predict it before its happening? How to prevent that failure? And how to minimize the failure rate? Most importantly find out the root causes behind the facture lead to the bad quality of soft-ware. Bad quality of software lead to enormous loss not only to the market share or financial loss but also loss in trust of consumer/customer. For the software quality experts, it is the main priority to produce such software that not only meet the needs of customer but also have high quality and backup quality environment. Proposed two step evaluation model for software quality assurance not only pin point the cause of problem effecting quality of software but also provide a solution environment to evaluate and erase that very problem. The goal of FIT4App is to discover

and avoid failure at the root step i.e. at development stage of mobile application.

FIT4Apps Theories

FIT4Apps state that the chances of failure exist after the launch of mobile application. This in turn give basis of H1 and H2 respectively.

T1-Less failure happens after mobile application launch

While using FIT4Apps approaches 85 percent less severe mobile specific failure recorded after mobile application has been launched. Comparing FIT4Apps approach to others traditional quality assurance methods, it is more efficient and defect preventive approach.

T2-After the launch of mobile application less failure exist overall

15 percent less failure occur after launch of mobile application while using FIT4App. T2 give basis to T3

T3- Effective detection and prevention of more mobile specific failure

75 percent more precise and effective prevention of mobile specific failure using FIT4App approach as compare to other software quality assurance approaches.

T4-Effective detection and prevention of more overall mobile specific failure

FIT4App lead to 5 percent prevention of overall failure during development of mobile application as compare to using other traditional quality assurance approaches. FIT4Apps implementation required no hardcore effort it is one of the effective and progressive quality assurance approach to prevent failure before it occur according to listed hypothesis. This rises question that can it be implemented without usage of extra sustainable effort. This clues to theory-5.

T5 – No extra need of sustainable effort then development effort

In one iteration only 1 percent effort is required with FIT4 then any other quality assurance approaches compared to overall system development. Along with T5, it is reported that the effort required for FIT4 is much lower than the effort saved due to earlier identification of failure and mobile application failure. Beside these software quality assurance approaches most common and easy approach to increase the software quality of mobile application use is Customer survey, Market survey, and current evolving trend of mobile software application. Software Quality assurance totally work on user feedback, what user need? How they want to use specific mobile app? What feature they want to introduce in their app? How performance, usability, portability, reliability affect them. It's all about customer/user. Getting feedback, conducting survey, arranging seminar, these approaches play a major part to improve the software quality assurance of mobile application. Following these approaches not only enhance software quality of mobile application but also help to find out the future trends in evaluation of mobile applications, as software application change gradually with the time passes on basis of customer demand so, software quality also changes gradually with the evaluation in the mobile software application.

IV. Conclusion

The purpose of this research paper is to address the software quality assurance of mobile software application. Research work is conducted by taking two points into account. First one is how change in mobile application effect its software quality? And second is to find out the defect on very basic level and remove it before it leads to any negative effect on the software quality. A software quality framework based object oriented matrix model is also proposed to ensure the software quality assurance of mobile application along with a tool based approach name PAPRIKA to detect software anti-pattern that lead to bad software.

Acknowledgment

This paper is written as a semester project by Sania Imran and Faiqa Mehboob under the supervision of Mehreen Sirshar

References

[1] D. L. Parnas. Software Aging. In Proc. of the 16th International Conference on Software Engineering, ICSE '94, pages 279–287, LosAlamitos, CA, USA, 1994. IEEE Computer Society Press.

[2] N. Moha, Y. Gu´eh´eneuc, L. Duchien, and A. Le Meur. DECOR: A Method for the Specification and Detection of Code and Design Smells. IEEE Transactions on Software Engineering, 36(1):20–36, Jan 2010

[3] Statista, Worldwide mobile app revenues in 2015, 2016 and 2020. The Statistics Portal. https://www.statista.com/statistics/269025/worldwidemobile-apprevenue-forecast/, 2016. (accessed 02/01/2017)

[4] S. Babin. Developing Software for Symbian OS 2nd Edition. Wiley, Hoboken, NJ, USA, 2007.

[5] R. Meier. Professional Android 2 Application Development. Wrox, Indianapolis, IN, USA, 2010.

[6] W.-M. Lee. Beginning iOS 4 Application Development. Wrox, Indianapolis, IN, USA, 2010.

[7] A. Wigley, D. Moth, and P. Foot. Microsoft Mobile Development Handbook. Microsoft Press, Redmond, WA, USA, 2007.

[8] D. R. Mohammad, S. Al-Momani, Y. M. Tashtoush and M. Alsmirat, "A Comparative Analysis of Quality Assurance Automated Testing Tools for Windows Mobile Applications," 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2019, pp. 0414-0419.

[9] K. Holl and V. Vieira, "Focused Quality Assurance of Mobile Applications: Evaluation of a Failure Pattern Classification," 2015 41st

Euromicro Conference on Software Engineering and Advanced Applications, Funchal, 2015, pp. 349-356.

[10] K. S. Arif and U. Ali, ”Mobile Application testing tools and their challenges: A comparative study,” 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2019, pp. 1-6.

[11] Y. Fan, X. Cao, J. Xu, S. Xu and H. Yang, ”High-Frequency Keywords to Predict Defects for Android Applications,” 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, 2018, pp. 442-447.

[12] T. Grønli and G. Ghinea, ”Meeting Quality Standards for Mobile Application Development in Businesses: A Framework for Cross-Platform Testing,” 2016 49th Hawaii International Conference on System Sciences (HICSS), Koloa, HI, 2016, pp. 5711-5720.

[13] T. Grønli and G. Ghinea, ”Meeting Quality Standards for Mobile Application Development in Businesses: A Framework for Cross-Platform Testing,” 2016 49th Hawaii International Conference on System Sciences (HICSS), Koloa, HI, 2016, pp. 5711-5720.

[14] P. K. Aggarwal, P. S. Grover and L. Ahuja, ”Locating Usability Critical Factors for Mobile Applications Using ELECTRE-TRI Method,” 2019 9th International Conference on Cloud Computing, Data Science Engineering (Confluence), Noida, India, 2019, pp. 596-600.

[15] Puneet Kumar Aggarwal, P.S. Grover, and Laxmi Ahuja, “Assessing Quality of Mobile Applications Based on a Hybrid MCDM Approach,” International Journal of Open Source Software and Processes, vol. 10, no. 3, pp. 51–63, 2019.

[16] H. Khalid, M. Nagappan and A. E. Hassan, ”Examining the Relationship between FindBugs Warnings and App Ratings,” in IEEE Software, vol. 33, no. 4, pp. 34-39, July-Aug. 2016.

[17] K. S. Arif and U. Ali, ”Mobile Application testing tools and their challenges: A comparative study,” 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2019, pp. 1-6.