



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

## A Critical Review on Software Metrics in Mobile Applications

Narinder Pal Singh, Monika, Nidhi

Research Student, Dept. of Information Technology, U.I.E.T, Panjab University, Chandigarh, India

Assistant Professor, U.I.E.T, Panjab University, Chandigarh, India

Assistant Professor, U.I.E.T, Panjab University, Chandigarh, India

**ABSTRACT:** Number of faults occurrence structures have been planned to recognize the entity in a software system that has the faults before the execution. Therefore, researchers are focusing on the identification of the number of fault presents in the software or identification of program modules of mobile applications that are most likely to contain faults. A lot of models have been developed using various techniques. A common approach is followed for software reliability prediction is based on software metrics. Detecting software faults early during development will definitely improve the reliability and quality in cost-effective way. So, this paper has presented the software metrics to solve the problem of prediction of defects in mobile applications.

**KEYWORDS:** Mobile Applications, Software Defects, Software Testing.

### I. INTRODUCTION

#### A. **MOBILE APPLICATIONS:**

Microsoft has developed the mobile application development that consists of system having mobile operations which is a descendent of a platform named window mobiles. Its development has begun in 2008 and a system was developed what gives user the best experience and fulfill user demands related to social media and the touch screen. All this work has started after Microsoft predicted mobile devices system. The authors had determined to restore the older interface on icon by incredible severely dissimilar. The novel interface depends on novel language of design named as 'Metro' that depends on typography and lessen on graphics. This design privileged the titles of flat colored. Text is always being the main center for the navigation purpose. Metro is described by Microsoft as below:

"Metro is our design language. The researchers called it Metro because it's modern and clean. It's fast and in motion. It's about content and typography. And it's entirely authentic".

Mobileapps should adapt themselves to different screen sizes. Interaction of user to desktop and web applications are limited to keyboard and mouse whereas mobile apps make user interaction as fluid as possible. Mobile testers have to ensure that the application performs efficiently when user interacts in different ways. Mobile apps depend on emulators and simulators which have their own capabilities and limitations. These apps are removed, installed and updated more as related to desktop applications.

Mobile apps are installed, removed or updated more frequently than desktop applications. Platform makes a difference to software development and maintenance process. There are many previous studies that propose defect prediction models based on static code metrics [1]. So, this paper will review various techniques to find out the defects.

#### B. **SOFTWARE:**

It is a platform where the user performs the duties. It is useful in number of submissions in the existing world. Below are some examples related to application software :

- Embedded System Firmware
- The layer that controls and coordinates the distributed system is middleware.
- Operating System
- Websites

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

The above mentioned applications need to be executed without any error. A better QOS should be provided to the user. The software should be checked properly like the working etc. It should be a primary job of the user to provide the service as per client's requirement and the reliable software delivery. Reliability of the software is termed as the software that provides less chances of software failure for a time given to the client in a precise situation. To estimate whether a system satisfies the requirements or not is the process of testing. This process can vary in three ways, i.e., real likely and the variability of the simulation results. In layman language, testing is defined as the process of executing a system for finding the gaps, flaws for describing the necessities or the needs [2].

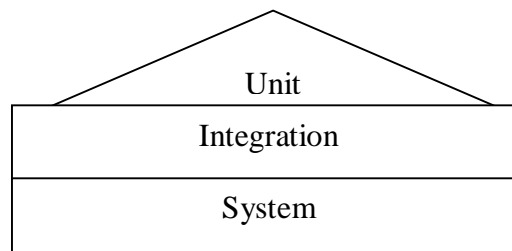


Figure.1 Software Testing Module

For the students, software engineering has become one of the necessities for boosting their knowledge and for their progression. The main objective is to develop the mechanism that installs and coordinates better with the requirements that already exists.. Programming reuse is one of the principle inspirations for CBSE. It concentrates on reusing and adjusting existing segments as opposed to creating them without any preparation. This lessens both the improvement cost and exertion and enhances the nature of the framework.

CBSE is extremely helpful to develop the competence except to create software dependable and to protect the eminence, it is essential to calculate the applications of the software. Here, to create software dependable plus reusable, it is extremely critical to ensure the importance at each stage earlier than testing toward to be conscious about the flaws at every stage of software.

## C. SOFTWARE EVOLUTION LAWS:

Lehman has given laws for software development. He partitioned the product into three unique classes:

- **S-category (static-category):** This is a software product, which lives up to expectations entirely as per characterized conditions and elucidations. The arrangement and the system to accomplish it, both are promptly seen before coding. The s-sort software is minimum subjected to changes henceforth this is the most straightforward of all. Case in point, calculator program designed for mathematical calculation [3].
- **P-category (practical-category):** This is a programming product with an assembly of techniques. This is characterized by precisely what strategies can do. In this software product, the details can be depicted yet the arrangement is not evident immediately. Case in point, gaming software.
- **E-category (embedded-category):** This software product meets expectations intently as the prerequisite of real-world environment. This software product has a high level of advancement as there are different changes in laws, charges and so forth in this present reality circumstances. Case in point, online trading software.

## II. SOFTWARE METRICS TO PREDICT FAULTS IN MOBILE APPLICATIONS

This phase can be distributed into three stages:

- Product metrics
- Process metrics
- Project metrics

Product metrics explains the distinctiveness of the creation like dimensions, difficulty, planned features, presentation, plus excellence stage. Process metrics improves the growth and preservation of the software. Like the efficiency of fault elimination in growth, the model of testing fault entrance, and the time of fixing the process. Project metrics explains the project uniqueness and implementation like the software developers, the patter of staff in cycle of software



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

life cycle and it also included the price, agenda, and output. A number of parameters fit in number of stages. For instance, project metrics and process metrics in the product metrics [6, 7, 8].

## A. **PRODUCT QUALITY METRICS:**

Following are the levels described by this phase :

- Mean Time To Failure
- Defect Density
- Customer Problems
- Customer Satisfaction

Number of bugs defines the quality of the intrinsic product in the software. It can even define as the time period the software runs before the software crashes. MTTF (mean time to failure) and DDR (Defect density rate) defines these criteria. For safety of the significant systems, MTTF parameters are used. For ex.: Avionics, airline traffic control measures, weapons. As suggested by the US government, the air traffic control mechanism cannot be occupied for more than three seconds for each year and civilian air lines has the availability of ten to nine hours [6,7], that should not be exceeded. In commercial software system, the defect density metric is normally used.

## B. **IN-PROCESS QUALITY METRICS:**

Process quality metrics has significant role. These metrics are less defined as compare to end product metrics. These metrics differentiate in number of software developers for the practices. To detect the defects in the machines while testing for some firm is in process quality metrics. Some of the organizations cover number of metrics in every phase of their development [9, 10].

## C. *Software Maintenance Metrics:*

While growths of a software invention is done and send to the market place then the maintenance phase of its life cycle begins. The faults arrival by instance gap and client difficulty calls by occasion gap are the “de facto” metrics. The defects or the problem can be determined with the growth process earlier than the maintenance process. More work cannot be done for editing the product in this phase. So, the “de facto” metrics cannot even help to give the maintenance of the software. The number of bugs should be removed as soon as possible a user can provide the best quality. The problems, although, cannot be overcome properly but assure the client to fulfill the needs.

The subsequent metrics are [11]:

- “Fix backlog and backlog management index”
- “Fix response time and fix responsiveness”[12]
- “Percent delinquent fixes”
- “Fix quality”

## III. LITERATURE SURVEY

Sharma and Jain [13] presented the WEKA method for classification. Proposed method gave the high accuracy in the result evaluation. It can be used in various fields like banking, medical area etc.

Kaur and Pallavi [14] proposed the utilization of machine learning algorithms for classification in software prediction.

Okutan and Yildiz [15] proposed review on the basis of accuracy parameter for fault prediction methods.

Geetanjali [16] proposed the various metrics for software fault prediction. In proposed method various methods has been utilized like LOC, lines of quality. Various algorithms has also been described in [17].

Garcia et al. [18] predicted blocking defects, which block other defects from being fixed. Furthermore, researchers have proposed to perform predictions at the change-level, focusing on predicting defect-inducing changes.



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

Zhang et al. [19] propose a contextaware rank transformation method to preprocess predictors and address the variations in their distributions, and showed that their cross-project models achieve performance that rivals within-project models.

Rohit Mahajan et al. [20] signifies that the software fault prediction model using BR technique gives improved accuracy as compare to (LM) Levenberg-Marquardt algorithm and (BPA) Back Propagation algorithm. The proposed work aims to find out the accuracy of Neural Network (proposed) with Bayesian Regularization (BR) techniques and the accuracy of BR method with LM (Levenberg Marquardt) with BPA (Back propagation )Algorithm is compared.

Tareq Aljaber et al [21] evaluated the mobile applications for health education. The needs and the output of number of stakeholders is achieved and gives improvement in software engineering with noval approach and the new enhanced ways are discovered for that software. Heuristic evaluation and usability evaluation parameters are used in the evaluation.

Cagatay Catal et al[22] provides a view on software on fault prediction used previously. It is being observed by the author that the percentage of the public datasets usage maximized considerably. But the machine learning usage percentage increases somehow from 2005.It is said that the use of class –level is less accepted and can be used more as compare to these days for predicting the errors in the designing phase of the software life cycle.

O. Í. Letychevskyi [23] has reviewed the model based testing methods. A noval method is proposed for testing by using the symbolic modeling and solves the variety of problems occurred by applying MRT method. The method depends on symbolic techniques. The method used predicate transformers for applying the symbolic methods and invariants generated.

Santosh S. Rathore et al [24] presented a technique by using a fault datasets for ten softwares in data repository “PROMISE” with genetic programming (GP). Various parameters (Error rate, recall and Completeness) are used for evaluating the performance of the research. For the number of faults predicted, the algorithm proposed (GP based) gives best results.

TABLE 1: Comparison Of Metrics

Author	Technique	Advantages	Disadvantages	Parameter
Sharma and Jain [13]	WEKA approach for comparative study of classification algorithm.	evolutionary algorithms is the best efficient tool in learning	Tests are carried out using the selected datasets. It is not universal	Accuracy
Kaur and Pallavi [14]	Data mining techniques for software defect prediction	Data mining methods are best to find defaults.	They found that higher support and higher confidence levels may not result in higher prediction accuracy.	Accuracy
Okutan and Yildiz [15]	Software defect prediction using Bayesian networks	Observe that there is a positive correlation between the number of developers (NOD) and the level of defectiveness.	Further investigation involving a greater number of projects is needed to confirm our findings.	Accuracy



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

Geetanjali [16]	Classification of defects in software using decision tree algorithm	The quality was assured using the quality metrics such as defect density and the accuracy was assured by sensitivity and specificity.	Automatically estimate the defect proneness of modules under development.	Lines of code and lack of coding quality.
R. Reddy & S. Kamath [17]	Bug classification: Feature extraction and comparison of event model using Naïve Bayes approach	improved accuracy for defects and reliability prediction type models	computation time can be significantly longer	Software maintenance speed
Garcia et al. [18]	Characterizing and predicting blocking bugs in open source projects	this models reduce the median time to identify a blocking bug by 3-18 days	Plan to include these blocking bugs in order to improve the accuracy of our model.	Prediction level
Zhang et al. [19]	Towards building a universal defect prediction model	evaluate the generalizability of the universal model by evaluating its performance using five external projects that are not hosted on SourceForge and GoogleCode	Not able to evaluate the feasibility of the universal model for commercial projects	Accuracy
RohitMahajan et al [20]	BR (Bayesian Regularisation) technique for finding software faults	reduce the cost of software testing which reduces the cost of the software project	Do not raise the accuracy level for finding the software faults at an early stage of software development life cycle	Accuracy
Tareq Aljaber et al[21]	mobile applications for health education.	Built an evaluation framework that depends on a combination of metrics (heuristic evaluation, user experiment and usability inspection).	A mobile health education application is not be improved by evaluating it in the early stage of software design	heuristic evaluation and usability evaluation.
Cagatay Catal [22]	systematic review of previous software fault prediction	usage percentage of public datasets increased significantly.	Did not Conduct more studies on fault prediction models using class-level metrics.	method-level
O. Ī. Letychevskiy[23]	Reviews existing methods of model-based testing.	solve some of existing MBT problems and the Generic Trace Generator based on the symbolic approach	Do not reduce an exhaustive search of large state spaces	-----



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

Santosh S. Rathore et al[24]	GP based fault prediction	Number of faults are predicted by the GP based results	-----	Error rate, Recall and completeness
------------------------------	---------------------------	--	-------	-------------------------------------

## IV. CONCLUSION & FUTURE WORK

In proposed paper, various software metrics for the detection of defects has been presented. This paper presented the utilization of machine learning algorithms in software fault prediction system. It has been found that the higher support and higher confidence levels may not result in higher prediction and accuracy. Also the number of bugs should be removed as soon as possible a user can provide the best quality. The problems, although, cannot be overcome properly but assure the client to fulfill the needs. The future scope lies in the usage of machine learning method that out performs than other classifiers for class level metrics. This method has better performance for method level metrics and useful in predicting fault prone classes for higher level of metrics (class). While research continues, practitioners and researchers may apply machine learning methods for constructing the model to predict faulty classes.

## REFERENCES

1. Chang C-P, Chu C-P, Yeh Y-F, "Integrating in-process software defect prediction with association mining to discover defect pattern", *InfSoftwTechnol*, vol. 51, pp. 375–84, 2009.
2. Khoshgoftaar TM, Seliya N, "The necessity of assuring quality in software measurement data", In Proceedings 10th International Symposium on Software Metrics, 2004.
3. Baojun Ma1 Karel Dejaeger Jan Vanthienen Bart Baesens, "Software Defect Prediction Based on Association Rule Classification", the 2010 International Conference on E-Business Intelligence, 2010.
4. D'Ambros M, Lanza M, Robbes R, "Evaluating defect prediction approaches: a benchmark and an extensive comparison", *Empirical SoftwEng*, vol.17, pp.531–77, 2012.
5. Beal CW, "Practical use of defect detection and prediction in the development and maintenance of software", In: Proceedings of the 4th international workshop on Predictor models in software engineering. Leipzig, Germany, ACM, pp. 45–6, 2008.
6. Qinbao Song, ZihanJia, Martin Shepperd, Shi Ying, and Jin Liu, "A General Software Defect-Proneness Prediction Framework", *IEEE Transactions on Software Engineering*, Vol. 37, no. 3, May/June 2011.
7. NaheedAzeem, ShaziaUsmani, "Analysis of Data Mining Based Software Defect Prediction Techniques", *Global Journal of Computer Science and Technology*, Vol. 11, 2011.
8. en.wikipedia.org/wiki/software\_bug.
9. B. Liu, Y. Ma, C.K. Wong, "Improving an association rule based classifier," In Proceedings of the Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases, 2000.
10. Stefan Lessmann, Student Member, Bart Baesens, Christophe Mues, and SwantjePietsch, "Benchmarking Classification Models for Software Defect Prediction: A Proposed Framework and Novel Findings", *IEEE Transactions on Software engineering*, Vol. 34, no. 4, JULY/AUGUST 2008.
11. J. Xu, D. Ho & L. F. Capretz, "An empirical study on the procedure to derive software quality estimation models", *International Journal of Computer Science & Information Technology (IJCSIT)*, AIRCC Digital Library, Vol. 2, Number 4, pp. 1-16, 2010.
12. S. Kumaresh& R. Baskaran, "Defect analysis and prevention for software process quality improvement", *International Journal of Computer Applications*, Vol. 8, Issue 7, pp. 42-47, 2011.
13. T. C. Sharma & M. Jain, "WEKA approach for comparative study of classification algorithm", *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 2, Issue 4, 7 pages 2013.
14. P. J. Kaur & Pallavi, "Data mining techniques for software defect prediction", *International Journal of Software and Web Sciences (IJSWS)*, Vol. 3, Issue 1, pp. 54-57, 2013.
15. A. Okutan& O. T. Yildiz, "Software defect prediction using Bayesian networks", *Empirical Software Engineering*, Vol. 19, pp. 154-181, 2014.
16. S. Adiu& N. Geethanjali, "Classification of defects in software using decision tree algorithm", *International Journal of Engineering Science and Technology (IJEST)*, Vol. 5, Issue 6, pp. 1332-1340, 2013.
17. S. J. Dommati, R. Agrawal, R. Reddy & S. Kamath, "Bug classification: Feature extraction and comparison of event model using Naïve Bayes approach", *International Conference on Recent Trends in Computer and Information Engineering (ICRTICIE/2012)*, pp. 8-12, 2012.
18. H. Valdivia Garcia and E. Shihab, "Characterizing and predicting blocking bugs in open source projects", In Proc. Int'l Working Conf. on Mining Software Repositories (MSR'14), pp.72–81, 2014.
19. F. Zhang, A. Mockus, I. Keivanloo, and Y. Zou, "Towards building a universal defect prediction model", In Proc. Int'l Working Conf. on Mining Software Repositories (MSR'14), pp.182–191, 2014.
20. RohitMahajan, Sunil Kumar Gupta, Sunil Kumar Gupta, "Design Of Software Fault Prediction Model Using BR Technique", Available online at [www.sciencedirect.com](http://www.sciencedirect.com) 849 – 858 1877-0509 © 2015 The Authors, Published by Elsevier B.V (http://creativecommons.org/licenses/by-nc-nd/4.0/). doi: 10.1016/j.procs.2015.02.154.
21. Tareq Aljaber, Neil Gordon, Chandrasekhar Kambhampati, Mike Brayshaw, "Tareq Aljaber, Neil Gordon, Chandrasekhar Kambhampati, Mike Brayshaw", *Science and Information Conference 2015 July 28-30, 2015 | London, UK*.



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

22. Cagatay Cataland Banu Diri, "A systematic review of software fault prediction studies", 0957-4174/\$ - see front matter \_ 2008 Elsevier Ltd. All rights reserved. doi:10.1016/j.eswa.2008.10.027.
23. O. I. Letychevskiy, "Paradigms of Model Based and Symbolic Testing of Software Systems", DOI 10.1007/s10559-015-9761-6 Cybernetics and Systems Analysis, Vol. 51, No. 5, September, 2015Springer Science +Buisness Media New York.
24. Santosh S. Rathore and Sandeep Kumar, " Predicting Number of faults in software System using genetic programming", 1877=0509,2015, Published by Elsevier,doi:10.1016/j.procs.2015.08.454.

## BIOGRAPHY

**Mr. Narinder Pal Singh** is currently a Master of Engineering student with the Information Technology, University Institute Of Engineering and Technology (UIET), Panjab University, Chandigarh. My research interest Include software engineering, mobile applications & software defects.

**Mrs. Monika** is currently an Assistant Professor at Information Technology Department, University Institute Of Engineering and Technology (UIET), Panjab University , Chandigarh. Her research interest include software engineering , software estimation & software engineering education.

**Ms. Nidhi** is currently an Assistant Professor at Information Technology Department, University Institute Of Engineering and Technology (UIET), Panjab University, Chandigarh. Her research interest include data mining & software engineering.