

---

**Recognition of Distinctive Reliability Leading Factors for Mobile Applications****Anupam**

SHUATS

Allahabad, India

anupamsharma243@gmail.com

**Dr. Tulika**

SHUATS

Allahabad, India

tulika.tulika@shiats.edu.in

**Article History:** Received:11 January 2021; Accepted: 27 February 2021; Published online: 5 April 2021

---

**Abstract** - Software reliability is a significant quality characteristic, and reliability models are often used to gauge and anticipate programming development. The quality of versatile apps conditions contrasts from that of PC and server conditions because of numerous elements, like the organization, energy, battery, and similarity. Assessing and anticipating versatile application dependability are genuine difficulties in light of the variety of the portable conditions in which the applications are utilized, and the absence of openly accessible deformity information. Also, bug reports are alternatively put together by end-clients. In the current research work, in view of the writing survey and specialist's assessment working in the field of versatile application advancement, 10 reliability leading factors and 14 sub-factors have been recognized that are fundamental for evaluating reliability of a portable applications.

---

**Keywords:** Mobile application, Reliability, Leading Factors, Quality Assurance.

---

#### I. INTRODUCTION

Software Reliability has gotten quite possibly the most significant client situated credits of programming quality [1]. It is of incredible significance to have viable ways to deal with create solid programming alongside quantitatively assessing the programming dependability [2]. Today by and large frameworks are programming based frameworks. The objective of the framework is to fulfill the client's requirements in the framework. Programming can't be seen or contacted, yet it is fundamental for the fruitful utilization of PCs. It is vital that the dependability of programming ought to be estimated and assessed. It has become a critical piece of numerous parts of society: home machines, media communications, autos, planes, shopping, evaluating, web educating, individual amusement, etc.

Programming Reliability is the vital errand for accomplishing the high dependability of any product industry. It applies the ascribes that are useful for accomplishing the unwavering quality and it center around measurements. IEEE defines reliability as "The ability of a system or component to perform its required functions under stated conditions for a specified period of time".

Reliability of software might be characterized as the likelihood of a thing to play out a necessary capacity under expressed conditions for a indicated timeframe. Programming Reliability is characterized as the likelihood of the disappointment free programming activity for a indicated timeframe in a predefined climate. Lack of quality of any item comes because of the disappointments or presence of issues in the framework. The untrustworthiness of programming is essentially because of bugs or configuration shortcomings in the product. It happens just when framework is being used and are not gone before by alerts.

Software reliability is an action for controlling and keeping up the cycles of the software development life cycle (SDLC) to create dependable programming. This action is utilized during the testing interaction until the cycle's leave measures are met. Moreover, programming dependability assists with keeping up and foresee the accuracy of the product [3]. Programming dependability designing was acquainted with help in dissecting and estimating the nature of programming applications. It presents the nature of the product running without delivering abandons [4]. Analysts and professionals have been improving programming unwavering quality models to survey the dependability of various kinds of programming.

Estimating and foreseeing the dependability of a mobile application are genuine difficulties because of the accompanying reasons. To begin with, the idea of versatile conditions is not the same as that of PC. Second, in versatile conditions, new functionalities and highlights are presented, like energy, organization, inconsistency, changed and restricted Graphical User Interface (GUI), interference, and notice, autonomicity which produce new

kinds of deformities [5]. Third, portable working frameworks and gadgets are jumpers. Fourth, the appeal for portable applications from clients has made the advancement interaction quick and the usefulness of versatile application more perplexing. At last, after a portable application is delivered, disappointments happen in cell phones. As well as testing, programming engineers incompletely depend on bug reports, which are alternatively put together by end-clients [16] [17] [18].

To survey programming dependability in portable applications, scientists are needed to invest more energy and exertion to assess the viability of programming unwavering quality. Considering the qualities of portable applications while estimating their product dependability will deliver more precise outcomes and examinations.

The current research work, focuses on deploying a good quality mobile application. To achieve the same, different leading factors and its related sub-factors get identified which influence the reliability of a mobile application in extreme levels. The rest of the paper is organized into different sections. The Section 2 discusses about the various models defined by various researchers in past and how it works. In Section 3 the identified leading factors and their sub-factors are discussed that plays the vital role in the assessment of reliability of mobile applications. Then in Section 4 conclusions are drawn from the work. Finally references at last.

## II. LITERATURE REVIEW

Because of the appeal of complex heterogeneous software's, programming dependability models have gotten more helpful to evaluate and foresee the rightness of the product [19] [20] [21]. Lyu [6] introduced programming unwavering quality models practically speaking to help specialists and professionals quantitatively address the attributes of the SDLC. Also, these models control designers and analyzers to comprehend and apply programming dependability strategies.

Tian et al. [7] assessed the dependability of web applications subsequent to deciding their imperfections and utilization. Also, the chance of upgrading web application dependability was investigated. The creator utilized the attributes of web applications as a base to order web deserts. The site responsibility was estimated and characterized at various levels and points of view, and joined with the disappointment information about the site to assess the operational unwavering quality. The experiment results demonstrate the viability and advantages of the creators' methodology.

Numerous SRGMs for assessing and anticipating the dependability of programming have been created and presented. In any case, a portion of these models show incorrect outcome. In this way, Huang et al. [8] looked into the strategic testing exertion work that can be utilized to portray the measure of testing exertion spent on programming testing.

Alansary and Tian [9] proposed a technique for measuring and foreseeing the dependability of SaaS. The creators broke down web worker log documents to separate disappointment information. The info area unwavering quality model was utilized to measure the operational dependability. SRGMs were utilized to gauge the development in SaaS unwavering quality.

Bokhary and Tian [10] proposed a framework for estimating the dependability of APIs. The creators followed a three-stage way to deal with gather accessible disappointment information, and afterward the API dependability was estimated. What's more, the creators presented a contextual investigation dependent on Google Map APIs and showed the adequacy and achievement of the proposed structure.

Perera [11] introduced a dependability expectation strategy to beat the irrelevant conventional dependability forecast techniques and convey more precise outcomes. Almering et al. [12] introduced an exact investigation to survey the dependability of programming and approve SRGMs during the mix and test stages. What's more, the capacity of the forecast model was contrasted with expectations by specialists. In addition, acquiring strong unwavering quality evaluation and expectation before programming discharge utilizing SRGMs was demonstrated to be conceivable

Ivanov et al. [13] introduced a correlation between the dependability of three working frameworks in portable conditions by applying SRGMs. Likewise, Meskini [14] assessed the unwavering quality of three versatile applications by applying SRGMs to disappointment information extricated from cell phones. Be that as it may, to

effectively survey and foresee programming unwavering quality, the attributes of versatile conditions should be thought of.

Getting to and differentiating quality leads a critical limit while performing quality assurance practices. Fulfilling buyer suspicions for emotional cell application has been an undertaking with extended prerequisites [15]. Hereafter, it anticipates that engineers should fabricate quality of things that a customer can use without any issues with unimportant undertakings. Subsequently, it expects engineers to expand quality of items that a client can use with no issues with negligible endeavors. In the current work, various reliability leading factors and its sub-factors get identified in order to design and deploy a good quality mobile application [22] [23] [24] [25].

### III. RELIABILITY LEADING FACTORS

In the current research work, in view of the writing survey and specialist’s assessment working in the field of versatile application advancement, 10 reliability leading factors and 16 sub-factors have been recognized that are fundamental for evaluating reliability of a portable applications. The characterized sub-factors are the estimation for the reliability leading factors. The 10 leading factors (RLF) includes program complexity of app, testing effort, customer experience, percentage of reused code, development team size, efficacy, design methodologies, testing tools, amount of programming effort, testing coverage [26] [27] [28] [29]. Every one of the factor distinguished assumes an essential part while estimating, regardless of whether a specific cell application comprise of with great reliability or not. Which thus characterizes as one of the subjective factor for assessing quality of the application [30] [31]. Along these lines, each factor decided is significant from the product application engineer's perspective just as with client's point of you. All the ten RLF are briefly described in Table 1.

Table 1: Reliability Leading Factors (RLF)

Reliability Leading Factors	Description
Program Complexity of App (RLF1)	Aside from figuring out what highlights go in an application, the intricacy of each factor likewise plays out a critical part in choosing the conveyance season of an application. Programming complexity is an approach to portray a particular arrangement of attributes of your code. These attributes all attention on how your code connects with different bits of code. The estimation of these attributes is the thing that decides the complexity.
Testing Effort (RLF2)	Testing exertion can be characterized as the quantity of testing cases produced, testing consumptions, or human years testing takes.
Customer Experience (RLF3)	This alludes to the individual scores and scrutinizes of the clients who previously utilizing the app and perceives different bugs at various levels.
Percentage of reused code (RLF4)	At the point when individuals build up some new programming products or when they update the old variant of their product items, they normally keep a portion of the module of the code which can be reused, and included some new ones.
Development Team Size (RLF5)	Team size affects the nature of programming items. Somebody accepts that huge group size will improve the nature of the product since there are more individuals engaged with the advancement cycle. In any case others guarantee that more modest however experienced improvement group will be better.
Efficacy (RLF6)	This refers to “how accurate consumer perceives the application without any error”.
Design Methodologies(RLF7)	Different plan systems for a similar programming may affect the nature of the final programming items. There are two sorts of plan procedures: structured plan and functional plan.
Testing Tools (RLF8)	There exist numerous different testing devices. These incorporate the product bundles analyzers use to convey out the testing undertakings. Different tools likewise give different quality and testing measures.
Amount of Programming Effort (RLF9)	The conscious programming effort might be viewed as effective for lessening the number of blunders made.
Testing Coverage (RLF10)	Test inclusion is defined as the level of the source code which is covered by the test cases.

The developers must focus on these identified additional factors identified specifically from the domain of mobile applications. Moreover, it's vital to describe how these factors must be assessed. For this, the described 14 sub-factors are identified further for evaluating reliability of a mobile application. This enables users to use a good quality mobile application [32] [33]. These 14 sub-factors are described in Table 2.

Table 2: Essential Reliability 14 Sub-Factors

RLF Sub- Factors	Description
Difficulty of Programming (DP)	This refers to ratio of the effort required for programming and the time required.
Programming Work Load(PWL)	During the product development, program load alludes to "work substance" for example, planning the development and a lot of work are the central point.
Programming Categories (PC)	The program classifications show framework intricacy. There are four program classes: working framework, correspondence control program, information base administration framework and Language processing.
Testing Methodologies (TM)	Diverse testing procedures contrastingly affect the quality of programming products. Good testing approaches may test more ways and need less time.
Level of Programming Technologies (LPT)	The programming innovations are characterized into four classifications: plan procedures, documentation strategies, programming methods and improvement PC access environment.
Development Working Standards (DWS)	Work standard is the standard the developing group needs to comply. This could be organization standard or gathering standard. Work standard shows products to be developed at each stage, plan archive design, plan depiction level and content, and the things to be checked in confirming the plan reports. Having these sorts of standards or not, affects the quality of programming items.
Testing Resource Allocation (TRA)	Testing resource allocation alludes to various plans to designate the testing assets, as far as testers, analyzers, and schedules of the testing exercises are concerned.
Testing Environment(TE)	To discover more mistakes during testing stage, testing environment should copy the operational environment. This can be characterized as the level of the similarity of testing and operational conditions.
Programming Skills (PS)	Programming abilities straightforwardly affects the reliability of programming items. It very well may be characterized as the average number of programming years of programming experience of developers.
Relationship between Design and Requirements (RDR)	Toward the finish of the planning stage, detail configuration is contrasted with necessities. Investigations are performed to confirm whether the capacities planned meet the prerequisites. Alterations can be made to eliminate the misconception between the clients and designers.
Software Application Security (SAS)	Application designers must be careful about security to protect client information. Working constantly to offer clients portable experience that don't bargain the information security is a tedious movement. Prior intensive security testing and not hurrying to deliver an application is an order to offer vulnerabilities free and reliable versatile application to the clients.
Software Application Design (SAD)	For making incredible and reliable applications, the need to have a powerful plan of application has begun to turn out to be progressively more significant.
Third Party Integration (TPI)	When the application needs to interconnect to different administrations, the need to incorporate with outsider APIs comes into picture. This requires additional opportunity to get familiar with the complexities of the outsider APIs and carry out them effectively. Then, it takes longer than anticipated time to assess the measure of work remembered for interfacing with outsider APIs and maintaining the reliability of an application.
Customer Support & Maintenance Services (CSMS)	Each application ought to have one client care segment since only one out of every odd client is educated. Furthermore, this client assistance ought to be completely working as well. The plan and execution as well as the client assistance is the thing that makes a fruitful application.

#### IV. CONCLUSION

Programming reliability is an action for controlling and keeping up the improvement measures with the objective of creating solid programming. Researchers and experts have been improving programming dependability models to survey the reliability of various kinds of programming products. Estimating and anticipating the reliability of a portable application are genuine challenges because of the contrasts between the nature of versatile applications, and PC and worker conditions. In the current research work, in view of the writing survey and specialist's assessment working in the field of versatile application advancement, 10 reliability leading factors and 14 sub-factors have been recognized that are fundamental for evaluating reliability of a portable applications. The outcomes showed that the reliability of versatile applications can be assessed and anticipated utilizing these identified factors and sub-factors. This empowers designers to assess and anticipate the dependability of versatile applications in more accurate manner so as to develop good quality mobile applications.

#### REFERENCES

1. Jiantao Pan, "Software Reliability", 18-849b Dependable Embedded Systems, CMU, 1999.
2. Musa, Iannino and Okumoto, "Software Reliability Engineering: Measurement, Prediction, Application.", Mc Graw Hill, 1987.
3. Michael R. Lyu , "Handbook of Software Reliability Engineering." McGraw-Hill publishing, 1995, ISBN 0- 07-039400-8.
4. P. K. Aggarwal, P.S. Grover, and L. Ahuja, "Evaluating Self-Management Features for Mobile Applications," International Journal of E-Services and Mobile Applications (IJESMA), 2018.
5. Kececioglu, Dimitri, Reliability Engineering Handbook. Volume 2, Prentice-Hall, 1991.
6. Lyu, M.R., et al. Handbook of Software Reliability Engineering. Vol. 222, IEEE Computer Society Press, Washington DC. 1996.
7. Tian, J., Rudraraju, S. and Li, Z., "Evaluating Web Software Reliability Based on Workload and Failure Data Extracted from Server Logs", IEEE Transactions on Software Engineering, 30, pp. 754-769, 2004.
8. Huang, C.-Y., Kuo, S.-Y. and Lyu, M.R. "An Assessment of Testing-Effort Dependent Software Reliability Growth Models", IEEE Transactions on Reliability, 56, pp. 198-211, 2007.
9. Alanssary, M.O. and Tian, J., "Measurement and Prediction of SaaS Reliability in the Cloud", 2016 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Vienna, Austria, pp. 123-130, 2016.
10. Bokhary, A., "Measuring Cloud Service Reliability by Weighted Defects over the Number of Clients as a Proxy for Usage", In Proceedings of the 32nd International Conference on Computers and Their Applications (CATA), Honolulu, HI, pp. 63-70, 2017.
11. Perera, U.D., "Reliability Index—A Method to Predict Failure Rate and Monitor Maturity of Mobile Phones", RAMS'06. Annual Reliability and Maintainability Symposium, Newport Beach, CA, pp. 234-238, 2006.
12. Almering, V., van Genuchten, M., Cloudt, G. and Sonnemans, P.J.M., "Using Software Reliability Growth Models in Practice. IEEE Software, 24, pp. 82-88, 2007.
13. Ivanov, V., Reznik, A. and Succi, G., "Comparing the Reliability of Software Systems: A Case Study on Mobile Operating Systems", Journal of Information Sciences, 423, pp. 398-411, 2018.
14. Meskini, S., Nassif, A.B. and Capretz, L.F., "Reliability Models Applied to Mobile Applications", 2013 IEEE 7<sup>th</sup> International Conference on Software Security and Reliability Companion, Gaithersburg, MD, 18-20 June 2013, pp. 155-162, 2013.
15. Goutam Kumar Saha, "Software Reliability Issues: Concept Map", IEEE Reliability Society 2009 Annual Technology Report.
16. P. K. Aggarwal, P.S. Grover, and L. Ahuja, "Security Aspect in Instant Mobile Messaging Applications," in Proceedings of IEEE International Conference on Recent Advances on Engineering, Technology and Computational Sciences (RAETCS), pp.1-5, 2018.
17. P. K. Aggarwal, P.S. Grover, and L. Ahuja, "Incorporating Autonomic Capability as Quality Attribute for Software Systems", In Proceedings of IEEE International Conference on Reliability, Infocom Technologies and Optimization, 2018.
18. P. Jain, A. Sharma, and L. Ahuja, "The Impact of Agile Software Development Process on the quality of Software Product", In Proceedings of the 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2018.
19. Song, K.Y, Chang, I.H. and Pham, H., "NHPP Software Reliability Model with Inflection Factor of the Fault Detection Rate Considering the Uncertainty of Software Operating Environments and Predictive Analysis", Symmetry, 11, 521, 2019.
20. P. K. Aggarwal, P.S. Grover, and L. Ahuja, "Exploring Quality Aspects of Smart Mobile Phones Applications," of Advanced Research in Dynamical and Control Systems (JARDCS), pp. 292-297, 2018.
21. Yamada, S., Ohba, M. and Osaki, S., "S-Shaped Reliability Growth Modeling for Software Error Detection", IEEE Transactions on Reliability, 32, pp. 475-484, 1983.
22. P. K. Aggarwal, P.S. Grover, and L. Ahuja, "Locating Usability Critical Factors for Mobile Applications Using ELECTRI-TRI Method", In Proceedings of IEEE International Conference on Confluence, pp. 597-600, 2019.
23. P. K. Aggarwal, P.S. Grover, and L. Ahuja, "Assessing Quality of Mobile Applications Based On a Hybrid MCDM Approach", International Journal of Open Source Software and Processes (IJOSSP), Vol. 10, Issue 3, Article 4, 2019.

24. P. K. Aggarwal, P.S. Grover, L. Ahuja, "A Performance Evaluation Model for Mobile Applications", In Proceedings of the International Conference on Internet of Things: Smart Innovation and Usages (IOT-SIU), 2019.
25. P. Jain, A. Sharma, L. Ahuja, "A Customized Quality Model for Software Quality Assurance in Agile Environment", International Journal of Information Technology and Web Engineering (IJITWE), Vol. 14, Issue 3, Article 4, pp. 64-77, 2019.
26. P. Jain, A. Sharma, and L. Ahuja, "ISM Based Identification of Quality Attributes for Agile Development", In Proceedings of the 5<sup>th</sup> International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), pp. 615-619, 2016.
27. P. Jain, A. Sharma, L. Ahuja, "The Model for Determining Weight Coefficients of Maintainability Criteria in Agile Software Development Process", In Proceedings of the International Conference on Internet of Things: Smart Innovation and Usages (IOT-SIU), 2019.
28. P. Jain and P.K. Aggarwal, "Mobile Phone Maintainability Prediction using MCDM Methodology", International Journal of Recent Technology and Engineering (IJRTE), Vol. 8, pp. 410-415, 2019.
29. P. Jain, A. Sharma, L. Ahuja, "Software Maintainability Estimation in Agile Software Development", International Journal of Open Source Software and Processes, Vol. 9, Issue 4, pp. 65-78, 2018.
30. P. Jain, L. Ahuja and A. Sharma, "Current State of the Research in Agile Quality Development", in Proceedings of the International Conference on Computing for Sustainable Global Development, pp. 1877-1879, 2016.
31. P. Jain, A. Sharma, P. K. Aggarwal, "Key Attributes for a Quality Mobile Application", in Proceedings of the International Conference on Cloud Computing, Data Science & Engineering, Confluence, pp. 50-54, 2020.
32. A. Sharma and Tulika, "Usability Assessment of Mobile Software Applications: A Systematic Review", Journal of Critical Reviews, Vol. 7, Issue 11, pp. 1865-1870, 2020.
33. A. Sharma and Tulika, "Identification of Distinctive Usability Critical Factors for Mobile Applications", Psychology and Education, Vol. 58, pp. 2649-2655, 2021.