

# Defining Mobile Quality Attributes Using Quality Function Deployment

Jane Kuria<sup>1</sup>

<sup>1</sup>Department of Information Technology, Dedan Kimathi University of Technology

DOI: 10.29322/IJSRP.9.06.2019.p90103

<http://dx.doi.org/10.29322/IJSRP.9.06.2019.p90103>

**Abstract-** Mobile app stores business model has made mobile apps available to millions of people anytime, anywhere. Developers develop apps that are available to users in short periods of time. Mobile applications are downloaded in their billions. Most of these apps fail. Quality of the mobile applications is the major factor. Users can now use an app and give feedback by way of rating and leaving comments. Negative comments affect the success of the mobile application. It is important for developers to act on the issues. This paper aims to define mobile quality attributes that are important based on analyzing users' reviews to understand the quality demanded. ISO/IEC 25010 is the current standard to measure the quality of mobile applications. Applying this standard as leverage we determined a relationship between what the customer quality demands are and the tool at hand to fulfil them using Quality Function Deployment. This should assist developers in prioritizing user requirements and their resources for mobile applications which should get to the market quickly.

**Index Terms-** Mobile app, Quality Function Deployment, quality attributes , app store.

## I. INTRODUCTION

Mobile applications are so prevalent in our society that Apple store slogan "there is an app for that" best describes [1] the growth. In 2019 Google Playstore has 2.1 million apps followed by Apple Store at 1.8 million according to Statistica.com. This growth can be attributed to the availability of smart device, ability of third parties to develop software and customers that are connected and on the go who need apps to improve their lives. Billions of apps are downloaded each year from these stores. A new business model has risen where development and distribution of mobile software is centralized [2]. Crowdsourcing has enable app stores to avoid the costs associated with development and marketing the app products, while providing a large variety of apps to their users at little or no risk. Quality of such a large number of applications is a challenge to guarantee with large numbers of users and developers scattered all over the world. In order to ensure quality apps the app stores have set up policy and publishing guidelines [3] that must be followed by all developers. Additionally the platform provides users with opportunity to give feedback for which developers are expected to act upon.

These paper proposes a way of measuring quality of mobile applications using customer requirements mined from app store reviews against a standard quality ISO 25010 through Quality Function Deployment.(QFD). QFD is a product development tool that focuses on translating customer requirements into design requirements that software engineers can [4] understand and implement. The customer requirement are captured from the mining users' reviews which are comments left in the app stores by users of a particular application. Review mining can be done manually or automatically. Negative reviews affect the users' view of is quality leading to less downloads and loss of revenue [5]. Other than increasing downloads, mining reviews and acting on the requests, can increase the rating of such mobile applications[6] [7] in the future. It assists in the maintenance and evolution of the app through crowdsourcing of the reviews.

The quality of mobile software conformance to specification and ability to meet the user needs can be view in three dimensions as proposed by [3] . These are the end user of the software product, the mobile environment where the software will execute and the regulations of the mobile app markets. The user needs are important in the perception of quality to a product. Their experience and overall satisfaction with the product plays a big role in determining the quality factors of a mobile software product. The mobile environment is different from ordinary software environment. There are constraints the device and constraints on network technology. Phones have smaller screens, limited power etc, while network technology like signal power, interoperability of different technologies affects the use of an application. The ability of a mobile application to adopt to these constraints will determine its quality. Mobile applications are now available in application markets like Google Playstore and IOS store. This app markets have regulations that have to be met in order for application to be hosted by them. They provide a source for software quality of the mobile software application.

This paper proposes a way of defining the most important quality attributes of mobile applications based on customer requirements mined from app stores and a standard quality tool that is at hand to fulfil them. Using QFD, we will prioritize quality attributes to help developers use their limited resources of software development more efficiently.

## II. LITERATURE REVIEW

### 2.1 Mobile Software Quality

Quality of mobile application is critical to its success in a very competitive market of app stores. Information, communication and leisure are some of the activities supported by mobile applications [8]. He continues to say that the user acceptance of a mobile software product depends on the users need and context and the perceived experience (Quality of experience) from using the product. He concluded that factors affecting QoE are, application interface design, application performance, phone features, battery efficiency, user routine, cost of application and connectivity we well as user's lifestyle.

In a qualitative survey of mobile software developers by [9], user experience that was seen as most relevant was for an app to be enjoyable, engaging, satisfying and helpful. They view the users of mobile applications to be different from those of desktop application. The mobile app users are fast and immediate response to their interactions. Information is expected to be available immediately in easy steps and interaction with other built in facilities like Bluetooth, GPS, Wi-Fi continue. Quality of mobile software as perceived by users is different from the technical quality [10]. This is both for internal and external quality of the mobile software. Mobile software applications quality can be seen as how the users performs and experiences the app under the constraints of the mobile application environment. Mobile software quality (both internal and external) is critical, user perceptions and expectations of an app are key in its economic success.

According to [11]. The lifespan of mobile applications is short. They are used for limited functionality. They add that complex functionalities in the software has to interface with to telephony, cameras, locations and other features unique to mobile devices. Usability issues can be raised by small screen displays, limited data input methods, context of use (mobile users that re affected by geographical and meteorological conditions), and connectivity that can change with location.. Battery and memory is still more.limited in mobile applications that software[12] . A mobile app needs to be transferable to many different platforms and their processing power is till limited.

Mobile application development process should not just be a variation of classical processes but will require techniques, methods and policies that address the new challenges it brings [13]. The development is happening in new unpredictable an unconventional settings and scenarios. It is difficult to predict what will work in terms of innovation customer satisfaction and return on investment.

The need to develop mobile software faster, users that are impatient and demanding and the constraints of the mobile environment, demand that developers prioritize user needs and put their limited resources where the will most effective.

### 2.2 Quality Function Deployment in software engineering

The use of QFD in software engineering is not a not new concept. In the mid-nineties the researchers [14][15] where looking in way of incorporating QFD in software development process. They referred to it as Software QFD and attempted to capture user requirements and associating with the Total Quality Management to rank their importance. Anang and others [16] apply QFD in Acceptance Driven Development (ATTD) to develop software that met user requirements and was maintainable. Test driven development is an approach of developing software, advocates for loosely couple codes for easy testability and maintainability.

Mobile software model is proposed by [17] by analyzing the published guidelines on quality by software quality mobile app store and ranking them. QFD is used to associate the demanded quality from app stores by the use of a quality standard.

## III. METHODOLOGY

The QFD to be used in this paper is in assisting the developers know which quality attributes matter most to the user. This is by gathering the user requirements that have been mined from user reviews. This will based on paper by [5] that mined user reviews from 20 popular app in Apple store. They manually analyzed 6390 reviews across 15 deferent categories. They found the most common complaints were as shown in Table 1.

Table 1: Customer Quality characteristics

Complaint type	Rank	Median (%)
Functional Error	1	26.68
Feature request	2	15.13
App crashing	3	10.51
Network problem	4	7.39
Interface design	5	3.44
Feature removal	6	2.73
Hidden cost	7	1.54
Compatibility	8	1.39
Privacy and Ethical	9	1.19
Unresponsive app	10	0.73
Uninteresting content	11	0.29
Resource heavy	12	0.28
<b>Not specific</b>		<b>13.25</b>

The twelve identified quality issues from mobile app users will be used as the customer requirements for our QFD. We hope to be able to associate these requirements with the current standard in place to measure them. The standard for this research will be ISO/IEC 25010, which is the current standard for quality for all software including mobile applications.

ISO/IEC 25010 is built from the ISO/IEC 25000 series of standards know as SQuaRE (System and Software Quality Requirements and Evaluation) in [18]. The standard is comprised of two models, the quality in use model and product quality model. The quality in use model has five quality characteristics which are further subdivided into sub characteristics. The product quality model comprises of eight characteristics that are further sub divided into sub characteristics.

Table 2: ISO/IEC 25010 software product characteristics

Product quality characteristics	Product quality sub-attributes
Functional Suitability	Functional completeness, Functional correctness, functional appropriateness
Performance efficiency	Time behaviour, resource utilization, Capacity
Compatibility	Co-existence, Interoperability
Usability	Appropriateness recognisability, learnability, operability, User error protection , user interface aesthetics, accessibility
Reliability	Maturity, availability, fault tolerance and recoverability
Security	Confidentiality, integrity, Non-repudiation, Authenticity, accountability
Maintainability	Modularity, reusability, analysability, modifiability, testability
Portability	Adaptability, Installability, replaceability

Table 3: ISO/IEC 25010 Quality in use characteristics

Quality in use characteristics	Quality in use sub-characteristics
Effectiveness	
Efficiency	
Satisfaction	Usefulness, Trust, Pleasure
Freedom from risk	Economic risk mitigation, Health and safety risk mitigation, environmental mitigation
Context coverage	Context completeness, flexibility

It is now possible to do a quantitative relationship between what the customers require from a mobile application and the ISO/IEC 25010 quality characteristics as outlined. We can then rank the relationship to provide us with the most important quality attributes to a mobile application customer. The methodology as outline by [3] is implemented in five steps as explained in the next section.

IV. RESULTS AND DISCUSSION

There are five major steps in implementing the QFD to ensure the quality requested by users is translated in the important quality characteristics of the final product.

Step 1. Customer Quality requirements: in this quality requirements of the customer are captured. This is done through analyzing customer reviews from apps stores. Twelve quality issues are reported as shown in table 1.. They are classified as “Customer Quality requirements” (CQR) which in table 1 are the complaint type

Step 2. Rating the importance of the customer quality attribute which is done by calculating the frequency of complaints or requests on particular quality attribute. This is captured by the paper written by [5]. There are 12 user complaints that are captured in Table 1.

Step 3. Standard quality: this describes the quality in terms of attributes that can be measured by a known standard. In this QFD the quality standard is provided by the ISO/IEC 25010 that sets standards for all types of software including mobile application software. Developers can use the thirteen quality characteristics and 42 quality sub-characteristics to measure the quality of the final product.

Step 4. Quality relationship matrix: A matrix is developed to determine the relationship between the customer quality requirement (CQR) and the quality standard (SQR). It measures the relationship between what the customer wants and standards available to ensure they get it. In QFD, a relationship value (RV) is calculated. The methodology define the relationships as strong that is assigned the value 9, moderate that is assigned the value 3 and weak that is assigned the value 1. Where there is no relationship no value is given. The relationship matrix is shown in table 3 and table 4.

Table 3: Relationship Matrix for software quality product standard

Customer quality characteristic	Importance	Functional suitability			Performance Efficiency			Compatibility		Usability				Reliability			Security			Portability				Maintainability								
		Functional completeness	Functional Correctness	Functional Appropriateness	Time Behaviour	Resource Utilization	Capacity	Co-existence	Interoperability	Appropriateness recognisability	Learnability	Operability	User Error protection	User Interface Aesthetics	Accessibility	Maturity	Availability	Fault Tolerance	Recoverability	Confidentiality	Integrity	Non-repudiation	Authenticity	Accountability	Modularity	Reusability	Analyzability	Modifiability	Testability	Adaptability	installability	Replaceability
Functional Error	26.68	Θ	β	Θ				α		Θ			α					β	Θ											Θ	α	
Feature request	15.13	β	Θ	Θ																												
App crashing	10.51	α	α	α	Θ	Θ	β		Θ		α	Θ	β	α				β	β									Θ	Θ	α		
Network problem	7.39				Θ	β		α													Θ	α	α									
Interface design	3.44									β	β	β	β	β	β	Θ	Θ															
Feature removal	2.73	Θ	Θ	β	α	α	α	Θ	α	Θ	α	α	Θ	α	α																Θ	
Hidden cost	1.54																		Θ	β	α	Θ	β									
Compatibility	1.39	Θ	Θ	Θ	Θ	Θ	β	β	β	Θ															α					α	α	
Privacy and Ethical	1.19												Θ						β	β	β	β	β									
Unresponsive app	0.73	β	β	β	β	β	β	Θ	Θ	Θ	α	Θ	Θ	α	α		α	α	Θ								α			Θ	Θ	α
Uninteresting cont	0.29	α	α	β																												
Resource heavy	0.28					β	β	Θ	Θ																							
<b>Absolute weight</b>		<b>229.1</b>	<b>315.24</b>	<b>173.86</b>	<b>67.17</b>	<b>114.03</b>	<b>118.92</b>	<b>50.41</b>	<b>57.19</b>	<b>125.55</b>	<b>44.93</b>	<b>67.41</b>	<b>215.97</b>	<b>75.18</b>	<b>13.73</b>	<b>10.32</b>	<b>0.73</b>	<b>99.77</b>	<b>348.45</b>	<b>95.37</b>	<b>24.57</b>	<b>34.42</b>	<b>22.72</b>	<b>32.24</b>	<b>0.28</b>	<b>1.39</b>	<b>0.84</b>	<b>1.39</b>	<b>32.92</b>	<b>115.99</b>	<b>41.61</b>	<b>8.92</b>
<b>Relative Weight</b>		<b>9.01</b>	<b>12.4</b>	<b>6.84</b>	<b>2.26</b>	<b>4.48</b>	<b>4.68</b>	<b>1.98</b>	<b>2.25</b>	<b>4.94</b>	<b>1.76</b>	<b>2.65</b>	<b>8.5</b>	<b>2.96</b>	<b>0.5</b>	<b>0.4</b>	<b>0</b>	<b>3.9</b>	<b>13.71</b>	<b>3.7</b>	<b>0.95</b>	<b>1.35</b>	<b>0.089</b>	<b>1.26</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1.29</b>	<b>6.1</b>	<b>1.63</b>	<b>0.35</b>

β-represents 9    Θ-represents 3    α-represents 1

Table 4: Relationship matrix for quality is use standard

Customer quality characteristic	Importance	Effectiveness		Satisfaction			Freedom from risk			Context coverage	
		Effectiveness	Efficiency	Usefulness	Trust	Pleasure	Economic risk mitigation	Health and safety risk mitigation	Environmental mitigation	Context completeness	Flexibility
Functional Error	26.68	β	Θ	β	Θ	β	α	α	α	Θ	Θ
Feature request	15.13	β	α	β		β				Θ	Θ
App crashing	10.51	β	β	β	β	β	α			Θ	Θ
Network problem	7.39	Θ	β	Θ		Θ	α	α	α	β	β
Interface design	3.44	Θ	β	β	Θ	β		α		Θ	Θ
Feature removal	2.73	β	Θ	β	Θ	Θ				Θ	α
Hidden cost	1.54			α	β		β				
Compatibility	1.39	Θ	β	Θ		Θ				Θ	Θ
Privacy and Ethical	1.19				β	Θ	Θ				
Unresponsive app	0.73	β	β	Θ	β	Θ	β			β	β
Uninteresting content	0.29			β	Θ	β				β	Θ
Resource heavy	0.28	Θ	β	Θ		Θ			β	β	Θ
Absolute weight		539.52	323.38	557.74	225.15	545.02	68.58	37.51	36.59	257.85	252.39
Relative weight		18.9	11.37	19.61	7.91	19.16	2.41	1.31	1.28	9.06	8.87

β-represents 9    Θ-represents 3    α-represents 1

Step 5: Standard Quality characteristic weight by importance. The association between the customer quality requirements and the standard quality to fulfil them can be calculated. First we calculate the Standard Quality Characteristic (SQR), which is given by the sum of multiplying the relationship value (RV) with the customer quality relative weight (CQRW).

$$SQR = \sum_{SQ=1}^n (RV * CQRW)$$

Afterward we calculate the standard quality relative weight (SQRW). This indicates the weight of the standard quality characteristics relative to other standard characteristics. This is then multiplied by 100.

$$SQRW = \frac{SQR}{\sum_{SQ=1}^n (SQR)}$$

The computed values for the standard quality characteristic and the standard quality relative weight are contained in Table nnn for the software product characteristics and in Table mm for the quality in use characteristics.

Ranking of quality characteristic is done to show the most important quality attributes as demanded by users. They are ranked from the most important to the least importance to make it easy to see what the users want at a glance. Figure 1 shows the software quality characteristics while figure 2 shows the quality in use quality characteristics.

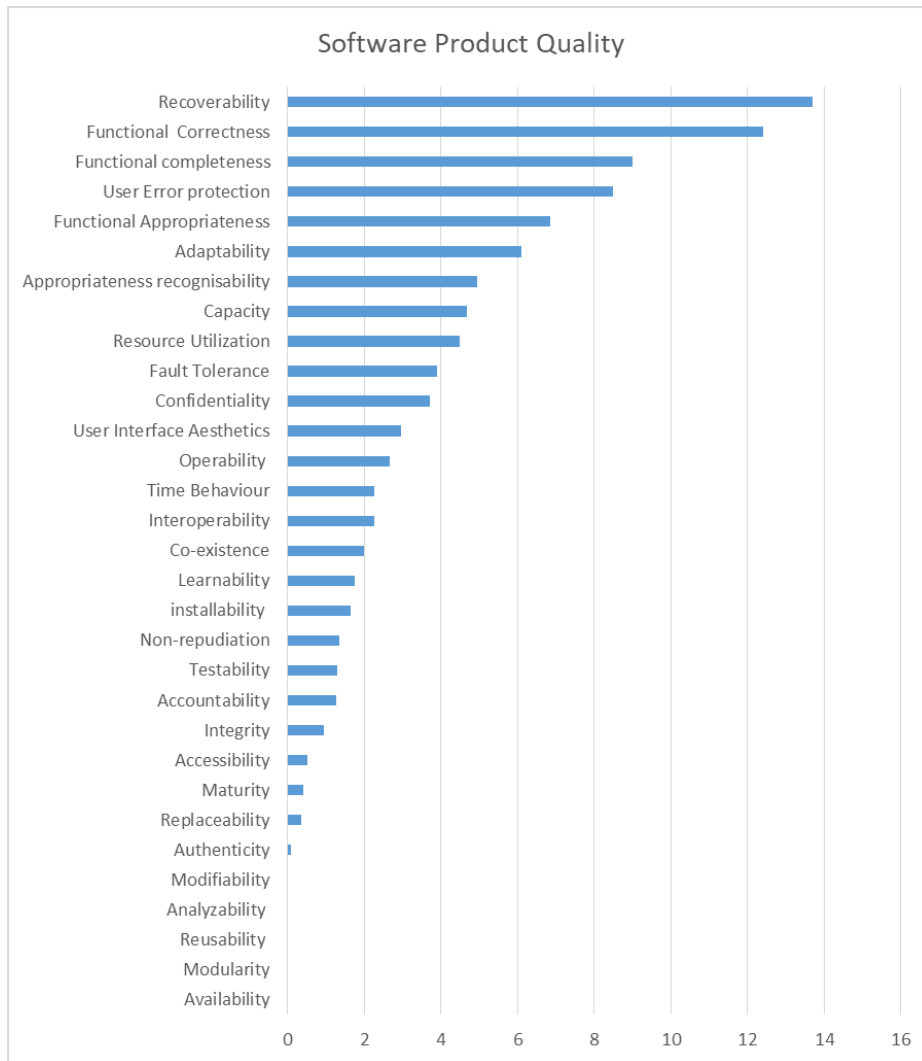


Figure 1: Ranking of quality attributes in software product standard

The software quality characteristics that customers demand apps have recoverability, functional correctness and completeness, user error protection and functional appropriateness as the most important quality attribute. Next tier of importance is adaptability, appropriateness recognizability, capacity, resource utilization, fault tolerance and confidentiality. Quality attributes of portability and maintainability are least important to users.



Figure 2: Ranking of quality attributes in quality of use standard

Quality in use characteristics are more important to users of mobile applications than software product quality characteristics in general. Customers demand mobile apps that are foremost useful, pleasurable to use and effective. Freedom from risk are the least important issues to mobile app users.

## V. CONCLUSION

In identifying the most important quality characteristics of mobile applications, functional suitability, usability and reliability are the most prioritized in the software product. Mobile app give least importance to portability and maintainability. In measuring the quality of the product while in use users prioritized usefulness, pleasure and effective. Least priority was freedom from risk. In developing software for mobile users, developers can use this knowledge in prioritizing resources when developing apps. Considering most mobile applications are developed quickly and updated frequently, this knowledge can help developers avoid negative reviews that affect the success of the mobile app in app markets.

## ACKNOWLEDGMENT

The authors would like to thank all participants.

## REFERENCES

- [1] T. Brockmann, S. Stieglitz, and A. Cvetkovic, "Association for Information Systems AIS Electronic Library (AISeL) Prevalent Business Models for the Apple App Store," 2015.
- [2] B. Bergvall-Kårebom and D. Howcroft, "The Apple business model: Crowdsourcing mobile applications," *Account. Forum*, vol. 37, no. 4, pp. 280–289, 2013.
- [3] L. Ricardo CORRAL VELÁZQUEZ and A. Luigi Frisiani, "A Software Assurance Model for Mobile Applications," Free University of Bozen.Bolzano, 2014.
- [4] L. C. Wood, C. Wang, H. Abdul-Rahman, and N. S. Jamal Abdul-Nasir, "Green hospital design: Integrating quality function deployment and end-user demands," *J. Clean. Prod.*, vol. 112, pp. 903–913, 2016.
- [5] H. Khalid, E. Shihab, M. Nagappan, and A. E. H. Member, "What Do Mobile App Users Complain About?," *Inst. Electr. Electronic Eng.*, pp. 1–7, 2015.
- [6] F. Palomba *et al.*, "Crowdsourcing user reviews to support the evolution of mobile apps," *J. Syst. Softw.*, vol. 137, pp. 143–162, 2018.
- [7] S. Panichella, A. Di Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, and H. C. Gall, "How can i improve my app? Classifying user reviews for software maintenance and evolution," *2015 IEEE 31st Int. Conf. Softw. Maint. Evol. ICSME 2015 - Proc.*, no. 2, pp. 281–290, 2015.
- [8] S. Ickin, K. Wac, M. Fiedler, L. Janowski, J.-H. Hong, and A. K. Dey, "Factors Influencing Quality of Experience of Commonly Used Mobile Applications," *IEEE Commun. Mag.*, vol. 50, no. 4, pp. 48–56, 2012.
- [9] R. Francese, C. Gravino, M. Risi, G. Scanniello, and G. Tortora, "Mobile App Development and Management: Results from a Qualitative Investigation," *2017 IEEE/ACM 4th Int. Conf. Mob. Softw. Eng. Syst.*, pp. 133–143, 2017.
- [10] A. Nitze and A. Schmietendorf, "A survey on mobile users' software quality perceptions and expectations," *2015 IEEE Eighth Int. Conf. Softw. Testing, Verif. Valid. Work.*, no. 0, pp. 1–2, 2015.
- [11] T. Vithani and A. Kumar, "Modeling the Mobile Application Development," 2014, vol. I, pp. 1–5.

- [12] V.-D. PAVALOIA, “Methodology Approaches Regarding Classic versus Mobile Enterprise Application Development,” *Inform. Econ.*, vol. 17, no. 2/2013, pp. 59–72, 2013.
- [13] P. Milano, A. Fuggetta, P. Milano, and E. Di Nitto, “Software Process,” pp. 1–12.
- [14] S. Haag, “Quality Function Deployment Usage in Software Development,” no. January 1996, 2014.
- [15] F. Authors, “Application of QFD to the,” 2006.
- [16] Y. Anang, “Collaborating Acceptance Test-Driven Development and QFD in Business System Software Development,” pp. 92–100, 2017.
- [17] L. Corral, A. Sillitti, and G. Succi, “from Publishing Policies of Mobile App Stores,” pp. 205–217, 2014.
- [18] International Organization for Standardization and International Electrotechnical Commission, “Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models : ISO/IEC 25010:2011,” pp. 18–20, 2011.