FUZZY LOGIC BASED QUANTIFICATION OF SOFTWARE QUALITY FACTORS FOR MOBILE APPLICATIONS

A
Thesis

Submitted to the

UNIVERSITY OF KOTA

in the Partial Fulfillment of the Requirements for the
Award of the Degree of

DOCTOR OF PHILOSOPHY

in

Computer Science
under the

Faculty of Science

Submitted by

Manish Mishra
Registration No.: RS/2305/16

Under the Supervision of
Prof. (Dr.) Reena Dadhich
Professor

Department of Computer Science & Informatics

UNIVERSITY OF KOTA
KOTA (RAJASTHAN)-324 005
INDIA

November, 2024

शोध निदेशालय
कोटा विश्वविद्यालय
एम.बी.एस. मार्ग, कोटा (राजस्थान)–३२४००५
फोननम्बर : ०७४४–२४७१०३७

Directorate of Research
University of Kota
MBS Marg, KOTA (Rajasthan)-324005
Phone No.: 0744-2471037

CERTIFICATE

I feel great pleasure in certifying that the Ph.D. thesis entitled **"Fuzzy Logic Based Quantification of Software Quality Factors for Mobile Applications"**submitted by Mr. **Manish Mishra** to the University of Kota in the partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy is based on the research work carried outunder my guidance.

He has completed the following requirements as per UGC Regulations and research ordinance of the University:

(a) Satisfactory Completion of the Ph.D. Course Work.
(b) Submission of Half Yearly Progress Reports.
(c) Fulfillment of residential requirement of the Research Centre (*Minimum 200 Days*).
(d) Presentation of research work before the Departmental Committee.
(e) Publication of at least one research paper in the referred research journal of national and international repute.
(f) Two paper presentations in the Conferences/ Seminars.

I recommend the submission of the Ph.D. thesis and certify that it is fit to be evaluated by the examiners.

Date:
Place:

(Prof. Reena Dadhich)
Research Supervisor
Professor & Head,
(Department of Computer Science & Informatics)
Dean, Faculty of Science
Director (Research)
University of Kota, Kota (Raj.)

शोध निदेशालय
कोटा विश्वविद्यालय
एम.बी.एस. मार्ग, कोटा (राजस्थान)–३२४००५
फोननम्बर : ०७४४–२४७१०३७

Directorate of Research
University of Kota
MBS Marg, KOTA (Rajasthan)-324005
Phone No.: 0744-2471037

DECLARATION

I, **Manish Mishra**, hereby certify that there search work presented in my Ph.D. thesis entitled **"Fuzzy Logic Based Quantification of Software Quality factors for Mobile Applications"**which is carried out by me under the supervision of **Prof. Reena Dadhich** and submitted in the partial fulfillment of the requirement for the award of the degree of Doctor of Philosophy of the University of Kota, represents my ideas in my own words and where others' ideas or words have been included in this thesis, I have adequately cited and referenced the original sources.

The work presented in this thesis has not been submitted elsewhere for the award of any degree or diploma from any other institution or university in India or abroad. I declare that I have adhered to all the principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission.

I understand that any violation of the above will cause for disciplinary action by the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Date: **(Manish Mishra)**
Place: Research Scholar

This is to certify that the above statement made by Mr. Manish Mishra (Registration Number:RS/2305/16) is correct to the best of my knowledge.

Date: **(Prof. Reena Dadhich)**
Place: Research Supervisor
Professor & Head,
(Department of Computer Science & Informatics)
Dean, Faculty of Science
Director (Research)
University of Kota, Kota (Raj.)

शोध निदेशालय
कोटा विश्वविद्यालय
एम.बी.एस. मार्ग, कोटा (राजस्थान)–३२४००५
फोननम्बर : ०७४४–२४७१०३७

Directorate of Research
University of Kota
MBS Marg, KOTA (Rajasthan)-324005
Phone No.: 0744-2471037

Anti-Plagiarism Certificate

It is certified that the Ph.D. thesis entitled "**Fuzzy Logic Based Quantification of Software Quality factors for Mobile Applications**" submitted by Mr. **Manish Mishra** has been examined with theanti-plagiarism tool.

We undertake that:

a. The thesis has significant new work/knowledge as compared already published or are under consideration to be published elsewhere. No sentence, equation, diagram, table, paragraph or section has been copied verbatim from previous work unless it is placed under quotation mark and duly referenced.

b. The work presented is original and own work of the author *i.e.* there is no plagiarism. No ideas, processes, results or words of others have beenpresented as the author's own work.

c. There is no fabrication of data or results which have been compiled and analyzed.

d. There is no falsification by manipulating research materials, equipment or processes, or changing or omitting data or results such that the research is not accurately represented in the research record.

e. The thesis has been checked by using Drillbit software and found within the limits as per UGC plagiarism policy and instructions issued from time to time.

Report is also enclosed along with this Ph.D. thesis.

Manish Mishra

Date:
Place:

(Prof. Reena Dadhich)
Research Supervisor
Professor & Head,
(Department of Computer Science & Informatics)
Dean, Faculty of Science
Director (Research)
University of Kota, Kota (Raj.)

DrillBit

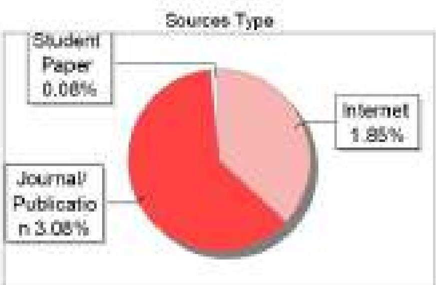
The Report is Generated by DrillBit Plagiarism Detection Software

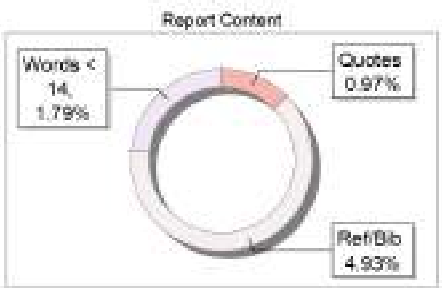
Submission Information

Author Name	Manish Mishra
Title	Fuzzy Logic Based Quantification of SQFs for Mobile Applications
Paper/Submission ID	2252138
Submitted by	profdadhich@uok.ac.in
Submission Date	2024-08-22 14:33:40
Total Pages, Total Words	198, 36906
Document type	Thesis

Result Information

Similarity **5 %**





Exclude Information

Quotes	Excluded
References/Bibliography	Not Excluded
Source: Excluded < 14 Words	Excluded
Excluded Source	**0 %**
Excluded Phrases	Excluded

Database Selection

Language	English
Student Papers	No
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

A Word of Gratitude

वर्णानामर्थसंघानांरसानांछन्दसामपि।
मङ्गलानांचकर्त्तारौवन्देवाणीविनायकौ॥1॥

बंदउगुरुपदकंजकृपासिंधुनररूपहरि।
महामोहतमपुंजजासुबचनरबिकरनिकर॥5॥

I would like to extend my sincere thanks
to my parents, wife, son, sister, brother
and family members.

ACKNOWLEDGEMENT

With due respect and pleasure, I express my feeling to Almighty God for devoting blessings and immense strength and power to accomplish the thesis honestly and successfully. I extend humble gratitude, acknowledgement and opportunity to do research work to my esteemed / honored supervisor Professor (Dr.) Reena Dadhich, Head, Department of Computer Science & Informatics, Director (Research) and Dean, Faculty of Science, University of Kota, Kota (Raj.) India, for her expert guidance, inspiration, continuous help, prompt support, encouragement, valuable comments, precious suggestions, motivation and innovative ideas throughout this work.

I gratefully thanks to Dr. Aparna Tripathi, Associate Professor, Department of CSE, Manipal University, Jaipur whose valuable suggestions and motivational advice pave a lot of creditability in programming & enhancing the overall tuning of the product.

This research would have been impossible without the support of my family members especially my wife and son, who always stood by me in my ups and downs, and always kept faith in my efficiencies.

I thank all faculty members, office staff members of Department of CSI UOK for their continuous support. My special thanks to Dr. Vipul Sharma (Dy. Registrar, Research Section, UOK), Mr. Chaman Tiwari, and other staff of Research Section, University of Kota who helped me during my research work.

I am extremely grateful to all my friends for their continuous support and encouragement.

Finally, I convey my special acknowledgments to the number of National and International researchers for referring my publications indexed in Scopus, Google Scholarand UGC Care.

Date: **(Manish Mishra)**

ABSTRACT

Quality of any product always is an asset for any organization. The implicit nature of the quality is subjective or qualitative. The answer of the question pertinent to any product "how is the product?" always comes up with "it is good" or "it is not up to the mark" and many other responses may be received as an answer to the above question. Quality of a software product is also subjective or qualitative in nature. Thus the question is "how can exactly the quality of software as a product be measured?" The answer to this question can be given only after performing the quantitative assessment. Thus, there is a requirement of particular kind of a framework which converts qualitative aspects of software product in to quantitative one, based on software quality factors (SQF) to obtain overall quality numerically.

A software development organization includes entities such as stakeholders, investors, quality managers, SQA team, developer, tester and β-Users with their different roles, but collectively, goal of all of them is to develop a quality software product. Broadly all entities may be divided in to four distinct categories (1) *decision makers* includes stakeholders, investors, quality managers (2) *developer, (3) tester* and (4) *β-Users*.

The proposed research focused upon quantifying the overall quality of a mobile application as a software product during its development. For this a fuzzy based mathematical framework is proposed which include software quality factors as per the domain of the mobile application by considering its limitations and quantify them using two inputs: *fuzzy rate* and *fuzzy weight.*

Mobile application has its own limitations as compare to desktop application. There are additional quality aspects in mobile application such as *customization, data-availability* and *test for mobility,* which are considered in proposed study. Three different perspectives of mobile application are taken in the proposed research work: (a) Developer (b) Tester and (c) β-User. Additionally, the implicit perspective will be offered by the decision makers with the assistance of proposed FW-MA (Fuzzy Weight for Mobile Application) algorithm based on pair-wise comparisons of quality factors and then appropriate fuzzy weights are assigned. The overall rating of three different perspectives which were analyzed using inputs fuzzy weight and fuzzy rate

ultimately resulted in an overall fuzzy rating. To measure the total crisp quality, defuzzification is performed. Defuzzification provides overall numeric value for quality of a mobile application. This numeric value will be the base for improvement in quality taken initially, if it's required.

The proposed fuzzy logic based mathematical framework MAQM-MA (Multi Attribute Quality Model for Mobile Application) which will be used for quantification of software quality factors of mobile application. The quality standard ISO/IEC 9126 and ISO/IEC-9241-11 are the basis for considering SQFs to design the proposed mathematical framework. The SQFs for developer and tester are considered from ISO/IEC-9126 and the SQFs for β-Users are considered from ISO/IEC-9241-11.

The main objectives of the proposed research work are as follows:

1. Understand the mobile application and allocate software quality factors (SQF) according to developers, testers and β-users.

2. Designing the five scale fuzzy based questionnaire for developer, tester and β-users to calculate the fuzzy rating for the metrics assigned by decision makers.

3. Relative priorities of software quality factors (SQFs) calculated by the proposed fuzzy based mathematical algorithm FW-MA for developer, tester and β-users. These relative priorities are the basis for decision makers to assign the appropriate fuzzy weights.

4. Evaluate the overall fuzzy rating of a mobile application under development from the perspective of developer, tester, β-user and their combined impact using the proposed fuzzy based mathematical framework MAQM-MA.

5. To evaluate the crisp quality, perform defuzzification using the centroid method for each fuzzy rating obtained from developer's, tester's, β-user's perspective and their combined impact, validated using the MATLAB Fuzzy logic tool.

Thus, the aim of this research is to design an appropriate fuzzy based mathematical framework for all mobile applications to measure the overall quality of mobile applications in accordance with their respective domains. In the field of quality management, this will be helpful to reduce the gap in between the quality of a mobile application and the customer's expectations.

This research study presents a novel application to perform pair wise comparison of SQFs in the form of fuzzy based mathematical algorithm named as FW-MA. The purpose of this algorithm is to determine the proper fuzzy weight for software quality factors of mobile applications. The proposed algorithm FW-MA will be of great help to the mobile application industry in quantitative evaluation of mobile application during the development process. The proposed MAQM-MA framework will also be implemented with a tool developed in Java Script and Node.js runtime. The MAQM-MA framework has also been tested on M-Commerce application as a case study.

The proposed fuzzy based mathematical framework MAQM-MA, increases investor and stakeholder confidence in mobile application development.

The results calculated with MAQM-MA for developer, tester, β-Users and the combined impact of these three perspectives are 75.89%, 62.80%, 58.75% and 62.65% respectively. After validation with the MATLAB fuzzy tool, the results obtained for developer, tester, β-Users and the combined impact of these three perspectives are 75.59%, 62.67%, 58.21% and 61.77% respectively. The results obtained with the proposed MAQM-MA and MATLAB fuzzy tool are almost identical, which shows that the proposed fuzzy based mathematical framework MAQM-MA serves its purpose.

CONTENTS / INDEX

LIST OF ABBREVIATIONS

Acronym	Full Form
CUI	Command User Interface
FW-MA	Fuzzy Weight for Mobile Application
FIS	Fuzzy Inference System
GUI	Graphical User Interface
GQM	Goal Question Metric
H	High
IEC	International Electro Technical Commission
ISO	International Organization for Standardization
L	Low
M	Medium
MAQM-MA	Multi Attribute Quality Model for Mobile Application
MD	Metrics for Developer
MT	Metrics for Tester
MβU	Metrics for Beta (β) User
QF	Quality Factor
QFD	Quality Factor for Developer
QFT	Quality Factor for Tester
QFβ	Quality Factor for Beta (β) User
SDLC	Software Development Life Cycle

SQA	Software Quality Assurance
SQF	Software Quality Factor
SQM	Software Quality Management
UEM-β	Usability Evaluation Model for Beta (β) User
VH	Very High
VL	Very Low

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

According to ISO/IEC 9000:2015 quality is a “degree to which a set of inherent characteristics of an object fulfils requirements”, where object referred as product, service, process, person, organization, system and resource [1].Quality is a subjective approach thus; it is possible that various individuals will have different methods to understand it. One of the most significant aspects of quality is that it is the factor that establishes the value of a product or service and it plays a significant role in achieving customer satisfaction, which in turn has an impact on the success of a company [2]. In order to guarantee the quality of the products or services that are being provided, there are a number of measures that are taken to ensure that the requirements and preferences of the consumer or client are satisfied. This is the ultimate goal of any company or organization.

A management approach which works for establishing the desired level of excellence or expected overall quality called as ***quality assurance***. Quality assurance is the study of building trust in meeting quality standards [3]. Quality can basically be described by how well a product, service, or event meets the needs of the customer. Quality of a software product is decided by the customer, not the developer. This is something that needs to be understood. Also, the idea of quality is not scientific but subjective. Quality of a software product varies in some different way; it is possible to measure quality of the software product as how well the factors of software quality meet the expected requirements of its users[4]. While deciding the quality, many attributes come into play, such as usefulness, speed, dependability, growth, security, maintainability, and customer happiness etc., but this subjective criterion must fulfill with the optimal or required number of attributes.

Software Quality Management, often known as SQM, defines as "systematic examination of the product or computer system, for example, when evaluating quality in use as part of quality assurance and quality control" [19]. Furthermore, it should work efficiently and without any errors. Also, the software should put security first, keeping private information safe and fighting against possible risks. Additionally, it should be easy to update, so that changes and bug fixing can go smoothly. Lastly, the software should be able to meet growth goals, which means it should be able to handle more users and data without affecting any of its functions.

The Software Development Life Cycle, also referred to as the SDLC, is a methodical procedure that involves strategic planning, the creation of software, testing, deployment, and maintenance. Integrating quality into every step of the software development lifecycle is necessary because it is so important at all stages of the software development process [5]. Development of software is an experimental rather than theoretical process [9]. To make sure the quality of software, development teams must come up with and strictly follow a thorough quality testing strategy. The suggested plan should include a complete set of methods, steps, and rules that make it easier to test and confirm the quality of software in a planned way. Another thing that the system should be able to perform is to find out possible risk areas, like issues related with security, maintainability or performance etc. This will also include an organized plan for dealing with such issues in more structured way [6].

Software quality assurance is a planned and systematic process that includes all the procedures which are required to offer sufficient confidence that an item or product corresponds to the technical standards. Quality assurance plans need to be looked at and updated on a regular basis to make sure it is meeting the goals of the software. High-quality software is essential for both business growth and customer happiness. To effectively achieve desired goals, it is necessary to develop the quality assurance plan and put it at the top priority.

A lot of different things, including modules and sub-modules that are built into the program itself, can affect the quality of a software product. To measure the quality of a software product, it can be divided into two groups: internal and external measurements [7]. Internal measurements decide about different features of program related to its developer. They include things like how reliable, maintainable, flexible, and efficient it is. External measurements, on the other hand, are how the user feels about the program and include its usefulness, functionality, and user-friendliness.Different software quality models were proposed by various researchers in [11, 12, 15, 22]. These models are proposed for generic software applications. Out of these models, ISO/IEC 9126 [15] is the most prominent standards. Sticking to well-known standards for quality, like ISO/IEC 9126 [15] and ISO/IEC-9241-11[28] are used to judge the quality of software. These standards give an organized way to judge the quality of software by looking at six main factors such as: functionality, efficiency, portability, maintainability,reliability and usability. It's important to note that the above quality factors play more important role to determine the quality of software as compared to other quality factors.This requires a systematic software quality model for developing and evaluating the software product [25].

The important aspects of research in the field of quality assessment of a software product under development are the “quantification of software quality factors”. Various researchers have made attempts to quantify the software quality criteria [10, 49, 50]. Costing is one of the aspects that must be considered during process of quantification [13]. While judging the quality of software, all of the necessary factors must be taken into account and carefully designed the overall quality model and then quantify.

1.1.1 Desktop Application vs. Mobile Application

A desktop application is a kind of computer program that may be run directly in the desktop environment's graphical user interface (GUI) or command user interface (CUI), often in its own window, and without using the web browser. When developing desktop software, developers often have a particular goal in mind. This may be anything: word processing, spreadsheet management, accounting, browsing the web, handling email, playing movies, viewing files, editing images etc. In contrast to mobile apps, desktop applications are often installed locally on the user's machine. Mobile applications, sometimes known as "mobile apps," are programs developed to use on portable devices like smartphones, tablets, and other similar gadgets. Google Play, Apple's App Store, and BlackBerry App World are examples out of the many app shops available for downloading programs for various mobile devices. A lot of mobile apps can also be downloaded from outside sources. There are many mobile apps that can perform variety of tasks. Some of these functions are entertainment through games, online learning, and online shopping.Mobile app creation is the methodical process of planning and making software programs that are especially made to work on mobile devices, such as Smartphone and tablets. Desktop apps are usually put as a user's computer and are used to do many things, like editing documents, working with spreadsheets, browsing the web, and more. Mobile apps, on the other hand, are made to work on small, movable devices like Smartphone and tablets. Most of the time, desktop programs have more functions and choices than their mobile versions. Desktop apps may use more system resources, which mean they might not work well on devices with limited access to those resources. Mobile apps, on the other hand, are designed to be easy to use and not require a lot of processing power [8]. This makes them perfect for people who are always on the go.

The following table (Table 1.1) illustrates the Comparison between mobile applications and desktop applications:

Table 1.1: Mobile Application vs. Desktop Application

Mobile Application	Desktop Application
1. Mobile applications often display a smaller allocation of resources and a more limited functional scope.	1. Desktop application has much higher configuration of resources than mobile application.
2. Mobile version of software can be missing certain feature due to limited resources.	2. Desktop version of the same mobile application has all features.
3. Reduction in the overall user experience due to small screen size.	3. Desktop application does not have such limitation.
4. There must be a tradeoff in between performancewith software and hardware overhead. The mismanagement may lead to slower loading times and performance	4. Desktop application does not have such limitation, due to capacity of software and hardware up gradation.

difficulties with some apps on mobile devices.	
5. Mobile apps have a stronger tendency to use resources than desktop programs do which results in a higher battery usage.	5. Desktop application does not have such limitation.
6. Mobile applications are more likely to be prone to security concerns than desktop applications. This is because mobile applications are more likely to be downloaded and installed on mobile devices with the help of internet.	6. Desktop application does not have such limitation, because they do not have issue of mobility.

The next important question is "how to understand the requirement of mobile application"? Researchers have their opinions to understand requirements of mobile users by surveying [57] and thus improve the quality of mobile app [55]. Extend the well defined quality standards to align with the mobile environment [53]. Quantification of software quality factor for mobile application is one of the challenges while considering its limitations [67].

1.2 RESEARCH PROBLEM

Quality of a product as software is a subjective or qualitative aspects. The overall quality depends upon many software quality factors. The qualitative aspects depend upon optimal number of software quality factors, carefully chosen by decision makers of any software development organization. When a software product is converted from a desktop program in to a mobile application,demand on quality increase even more. In situations where time is of the essence, mobile apps are often used, and consumers expect them to work reliably and consistently. There are so many quality standards that are taken to be considered which may customized in such a way that design a generic software quality model for mobile application. This research study considers quality standardsISO/IEC 9126 [14] and ISO/IEC 9241-11 [28], two key standards used to evaluate the overall quality of mobile application under development.

Mobile devices come with their own set of constraints, such as power issues, limited resources, connection, and so on; quality managers understand these issues in conjunction with other software quality concerns. This will provide a basis which formulates overall qualitative aspects. The goal is to numerically measure the qualitative aspects of a given mobile application, which can then be further used in variousways; to compare the developed prototype of a mobile application with another existing application. This is only possible if we convert the qualitative aspectsinto a quantitative evaluation, which requires a suitable mathematical framework.

There are a number of difficulties associated with assessing theoverall quality of mobile applications, some of them are:

1. Qualitative assessment: The components that determine quality are often subjective and can be difficult to measure. For this reason, it is very challenging to design qualitative aspects for the respective area.

2. Customization of software quality factors (SQFs): Collecting software quality factors according to the mobile application domain can be challenging. Adapting the software quality factors to the respective mobile application is another challenge.

3. Designing software quality metrics: Themethodology behind the design of software quality metrics or questionnaires can be challenging.

4. Data: To be considered valid, quality measures must be based on data that is both accurate and relevant, which can be difficult to obtain.

5. Assessment: For quality assessment to be successful, it must be quantifiable. This requiresa system of quality standards that is recognized by the industry.

6. Designing a mathematical framework: Designing an optimized mathematical frameworkcan be a challenge.

7. Validation: In order to guarantee that the quality evaluation is correct and trustworthy, it is necessary to verify it. The fact that this situation calls for a substantial quantity of testing and validation might make it challenging to achieve what is needed.

In view of the above challenges, it is therefore necessary to design a suitable framework for mobile applications in order to access the software quality factors.

1.3 RESEARCH OBJECTIVES

The main objectives of the proposed research work are as follows:

1. Understand the mobile application and allocate software quality factors (SQF) according to developers, testers and β-users. The SQFs for developer and tester are considered from ISO/IEC-9126 and the SQFs for β-Users are considered from ISO/IEC-9241-11.

2. Designing the five scale fuzzy based questionnaire for developer, tester and β-users to calculate the fuzzy rating for the metrics assigned by decision makers.

3. Relative priorities of software quality factors (SQFs) calculated by the proposed fuzzy based mathematical algorithm FW-MA for developer, tester and β-users. These relative priorities are the basis for decision makers to assign the fuzzy weights.

4. Evaluate the overall fuzzy rating of a mobile application under development from the perspective of developer, tester, β-user and their combined impact using the proposed fuzzy based framework MAQM-MA.

5. To evaluate the crisp quality, perform defuzzification using the centroid method for each fuzzy rating obtained from developer's, tester's, β-user's perspective and their combined impact. The results are validated using the MATLAB Fuzzy logic tool.

Thus the aim of this research is to design an appropriate mathematical framework for all mobile applications that contains improved quality criteria and measures the overall quality of mobile applications in accordance with their respective domains. In line with quality management, this will help to reduce the gap in between overall quality of mobile application and the customer's expectations.

1.4 MOTIVATION

Mobile applications have become an essential part of the digital lifestyle that everyone lives today. They are used extensively to provide support in a variety of facets ofpersonal and professional life. However, the quality of mobile applications is a qualitative issue, but for analysis it is necessary to represent the quality in numerical form.To solve this problem, a mathematical framework is required that quantifies the overall qualitative aspect of mobile applications. Sucha model will enable developers, tester and β-user's to make timely informed decisions about the overall quality of mobile applications, so that decision makers can take corrective actions that will minimize the gap between the actual customer requirements and the delivered mobile application.

The motivation to develop a fuzzy based mathematical framework to access the overall quality of mobile apps might originate from a number of various sources. The fact that the framework is able to provide numerical ratingsfor the quality of mobile apps is one of the main motivations for the current research efforts. So far, there is no mathematicalframework that could help developers and testers who need to access the quality of their apps and make adjustments. This leads to the fact that there is a significant need for such a framework. Evaluation in numeric formcan also be used to facilitate comparisons between different applications and the identification of areas for improvement.

1.5 RESEARCH METHODOLOGY

This research is conducted using qualitative and quantitative research methodologies. Fuzzy questionnaire based on five scales for developers, testers and β-users to calculate the fuzzy rating for the metrics as designed. A survey was conducted on prototype of M-Commerce application and the questionnaire was filled by developer, tester and β-users to determine the fuzzy rating for each metric. These fuzzy ratings along with the fuzzy weight provided by decision maker, serve as input to a proposed fuzzy based mathematical framework. This framework calculates the overall fuzzy and crisp quality according to developer, tester, β-users and their combined impact. Thus, the overall quality of the mobile application from these three perspectivesand their combined effect is quantitatively evaluated using the proposed framework. Fig. 1.1 shows the workflow of the proposed research work.

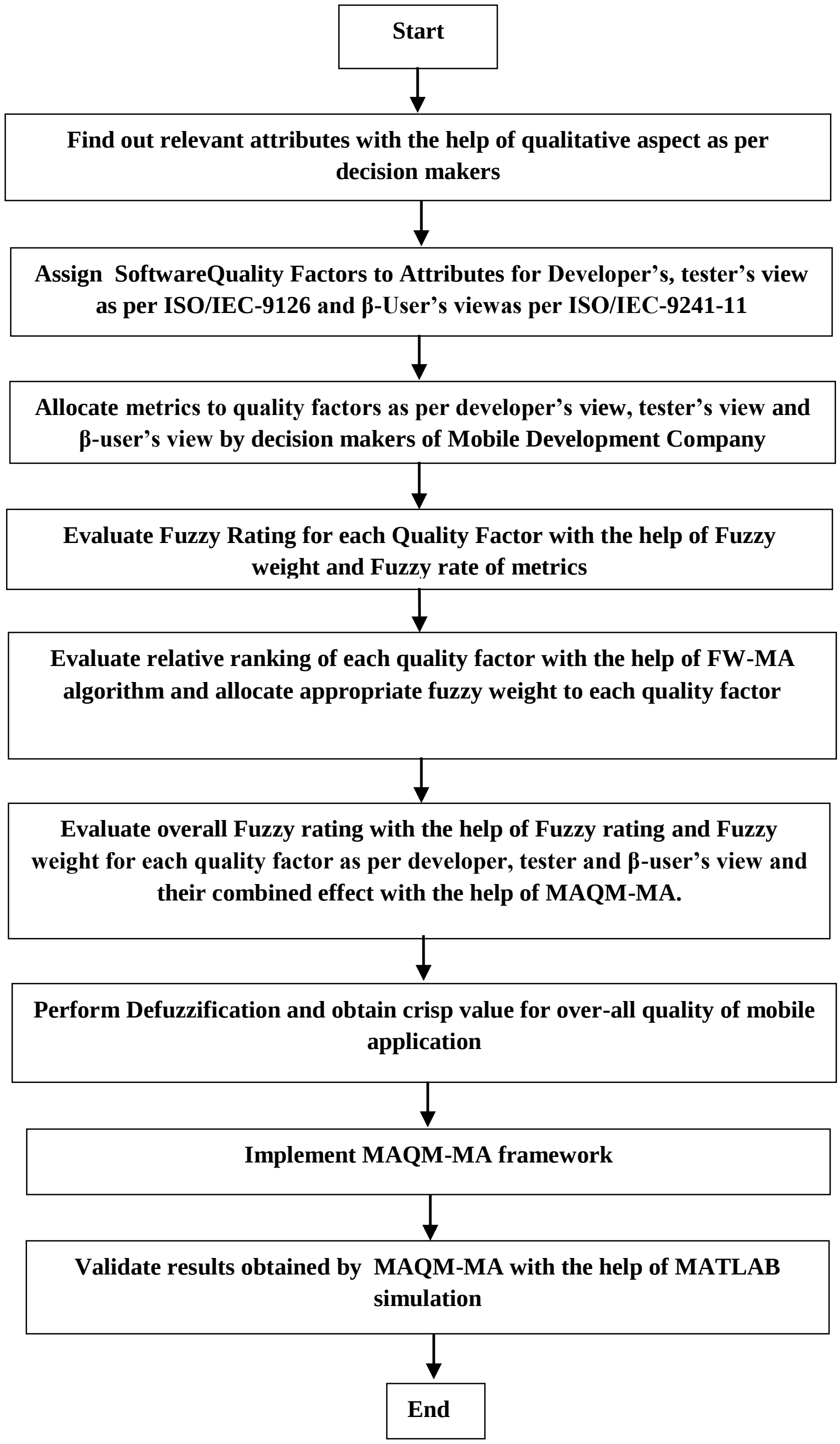


Fig 1.1:Workflow of the Proposed Research Work

1.6 ORGANIZATION OF THESIS

Thisproposed research work is divided into nine chapters, listed below:

Chapter 1 Introduction

In this chapter, the research problem is precisely defined along with the research objectives. This chapter ends with the research methodology and workflow.

Chapter 2 Review of Literature

This chapter provides detailed information about the research studies that have been conducted in the area of quantifying software quality factor for mobile applications.

Chapter 3 Software Quality Factors for Mobile Application

This chapter explains the strong correlation between the software quality factors for mobile application and the metrics and concludes with the establishment of usability framework for a mobile game app as a case study, which is further utilized for the purpose of quantification.

Chapter 4 Fuzzy Logic based Quantification for Mobile Application

This chapter explains the quantification process computed using fuzzy logic and performs the fuzzification and defuzzification. In this chapter, fuzzy logic is used for converting qualitative assessments into quantitative ones.

Chapter 5 Proposed Mathematical Framework MAQM-MA: Multi-attribute quality model for Mobile application

In this chapter, a fuzzy based mathematical framework MAQM-MA is proposed. This chapter starts with the concept and design process of MAQM-MA, which includes another fuzzy based mathematical algorithm FW-MA used to assign fuzzy weights depending on the decision maker.

Chapter 6 Quantification of Quality factors using MAQM-MA Framework

In this chapter, a case study on M-Commerce is used to quantify a fuzzy-based mathematical framework MAQM-MA together with the FW-MA algorithm. The case study evaluates the overall fuzzy and crisp quality from three perspectives. The quantification helps in evaluating the quality of mobile application throughout the development of the mobile application.

Chapter 7 Implementation of MAQM-MA Framework

This chapter discussed software modules for implementing the fuzzy-based mathematical framework MAQM-MA using Java Script and Node.js.

Chapter 8 Validation of MAQM-MA Framework

This chapter explains about data validation through MATLAB Fuzzy Logic Tool.

Chapter 9 Conclusion & Future Scope

This chapter concludes about various aspects of MAQM-MA mathematical framework along with the discussion about a novel algorithm FW-MA, which evaluates relative order with in quality factors and provides feedback to decision makers for allocation of appropriate fuzzy weight.

CHAPTER 2

REVIEW OF LITERATURE

2.1 INTRODUCTION

There is a clear correlation between the quality of any product and their level of trustworthiness. An excellent piece of software has a significant bearing on the degree to which people have faith in it. There is a correlation between the software's reputation and the likelihood that people will trust it and utilize it. On the other hand, if the program has a poor reputation, users could be less inclined to trust it and utilize it. The quality of the software product is critical for establishing confidence between users and developers, as well as for enhancing the level of happiness experienced by customers. The user experience is improved when quality software products are more dependable, simpler to use, and include fewer problems. All of these factors contribute to a satisfactory user experience. In addition, software products of high quality often have a greater number of features and better documentation, which makes it simpler for customers to comprehend how to make use of the product effectively. An increase in user trust, which in turn leads to increased customer satisfaction and improved customer loyalty, is one of the overall benefits of superior software solutions.

For the purpose of review of existing literature the current research activity is divided into four distinct categories:

(a) Software Quality Factors and Software Metrics.

(b) Application of soft-computing to quantify software quality Factors.

(c) Application of MCDA for quantification.

(d) Quantification of Software Quality Factors for mobile application.

2.2 SOFTWARE QUALITY FACTORS AND SOFTWARE METRICS

Software metrics that are used to determine the degree of quality produced by a software product are referred to as software quality factors. In addition to functionality, usability, dependability, maintainability, performance, security, scalability, and portability, quality considerations may also contain other similar characteristics. It is important for a developer or quality assurance team to do an analysis of each of these factors when assessing the quality of software. Software metrics represents software quality factor, hence software metrics are carefully designed to enhance the impact of software quality factor.

V. R. Basili,[9] explain that software development activity should be considered asexperimental rather than theoretical, and this paper presents anew project-specific model that can be better understoodwith the aid of knowledge, process, and product. Thismodel is intended primarily for educational and researchpurposes.

V. R. Basili,[10]explain that anassessment modelshouldcontainmetricsthatquantifythequalityoftheengineeringprocess. There are manyvariables which influencestheoverallqualityofsoftwareprograms.Thus,thegoalofquantification is to improve the entire quality of software as a product.

J.A. McCallet. al [11], proposed an approach to establish quantifiable criteria to produce reliable and error-free software system for Air force system. All qualitative terms defined as a subjective term, due to which it is not possible to find out overall quality of software product. Thus there must be some method to quantify qualitative terms in order to judge the overall quality of software product. This paper suggests the following steps:

1. Explore set of quality factors
2. Define criteria for each quality factors.
3. Define matrices for each criterion
4. Find overall rating.
5. Validate matrices with data of air force system.

Transfer result in to guide lines or standard which is used in air force.

R. G. Dromey, [12] defines a comprehensive framework for assessing the set of defects in a programming language to realize degree of defects and enhances improvements and providing feedback on the quality of the product. The model can be applied to existing systems or to new systems, and can be adapted to specific situations. As a result, it can provide an effective means of improving quality of software product. The paper provides an effective approach in the following ways:

1. To implement quality factors in to a programming language.
2. This model also provides an assistance to explore the area of defects.
3. This will help to provide set of defects for any particular language.
4. Anyone can realize the degree of defects.

S. A. Slaughteret. al [13], is to visualize software quality improvement as an investment. There must be trade-off in between investment for software quality and total budget, do not invest too much for software quality. Software companies must ensure about finance involved vs. software quality improvement efforts. The paper discussed the following points:

1. Analyze the trade-off in between investment and improvement of software quality.
2. Paper raises an important issue regarding quality improvement process such as avoidance of design reviews and code inspection.

Paper proposed two parameters COSQ (cost of software quality) and ROSQ (return on software quality) and evaluate them to justify investment and quality of software Improvement of software quality viewed as an investment, thus it is important to justify each software quality improvement effort and reduce the total cost on software quality.

L.Buglione and A. Abran[14], analyzes importance in order to design a mathematical tool for quality measurement of software based upon three dimensions:

1. Economic dimension as manager's view
2. Social dimension as user's view
3. Technical dimension as developer's view

Paper proposed a mathematical model to evaluate overall quality factor (QF) from three viewpoints. This paper collects data with the help of checklist based upon ISO/IEC-9126.

ISO/IEC 9126-1:2001quality model [15] defines six quality characteristics for software products: functionality, reliability, usability, efficiency, maintainability and portability. The model provides an effective way to assess software product quality and can be used as a basis for developing quality assurance programs. It is important to note that the quality model does not provide a definitive assessment of software product quality; instead, it is intended to serve as a reference for assessing software quality relative to the goals and objectives of the software product.

O.Lamouchiet. al [16], propose a design framework which evaluates quality of IS in an organization. This paper proposed IS quality model as set of properties such as reliability, maintainability etc. arranged in the form of hierarchy. The paper describes an approach to evaluate overall IS quality model according to top-down approach, where top represents most general properties understood by variety of people but difficult to measure directly and bottom includes metrics to quantify overall quality model.

Y. Jianget. al [17], proposed a tool for software quality prediction with the help of design and code metrics, it is expected to explore faults in software product as early as possible in software development life cycle. Design, code and requirement metrics have been used for predicting fault prone modules. This paper analyzes thirteen dataset from NASA metrics data program based upon code and design metrics. This paper also conclude that models that use combination of design and code level perform better than those models who use either of code or design metric set.

Y. Kanellopouloset. al [18], come up with a methodology for the evaluation of source code. The proposed methodology draws upon the ISO/IEC quality framework, which has been widely accepted by the global community. This methodology comprises four distinct stages. In the primary stage, specific ISO/IEC 9126 quality standard characteristics are selected, which depends upon source code internal quality and static behavior. The second stage involves identifying the source code attributes that directly influence characteristics and sub-characteristics selected in the first stage. These attributes should enclose all aspects of a software system, regardless of the paradigm (structured or object-oriented) employed in its development. Next, the methodology demands to choose metrics that are suitable for the evaluation of structured or object-oriented systems. The final stage involves assigning weights to source code attributes, which reflect their significance in evaluating ISO/IEC-9126 characteristics and sub-characteristics.

SQuaRE system [19] as described in the ISO/IEC 25022:2016 standard, provide a comprehensive framework for software and system quality requirements and evaluation. It covers all aspects of quality from design, implementation, and testing to deployment and maintenance, and provides a solid foundation for organizations to use when creating and evaluating software quality. The SQuaRE system and software quality models can help organizations to focus on quality improvement, ensure compliance with industry standards, and continually improve the quality of their software systems.

Ming-Chang Lee, [20] proposed a designed framework of software quality criteria and define software metrics and applies in software life cycle with software quality assurance. This paper proposes one or more QA quality measure matrices focus on ensuring the quality of the process and the resulting product. This paper discussed software quality matrices which includes RADC's methodology, project metric, reliability matrices, readability matrices, customer problem matrices, test product, process matrices etc. Overall this paper improves software product, processes and customer services with the help of appropriate set of quality matrices.

R. M.Ammadet.al [21], proposed that software quality factors and metrics are essential for improvement of SQA process. This paper explores the most effective factors and matrices which are helpful to enhance the degree of SQA process. This paper identifies important factors and matrices based on literature survey and study of different research works. This study can be treated as an asset for students, scholars, researchers and software developers.

B. W. Boehmet. al [22], proposes a conceptual framework and some key points in the analysis of characteristics of software quality. This paper focuses on:

1. Attention to characteristics of software quality.
2. It is important because current state of art imposes specific limitation.
3. A well defined hierarchy must be defined, from higher level structure to lower level matrices.
4. Software quality evaluation matrices have been defined and classified.
5. Realize software life cycle which have significant impact on software quality.

This paper states that it provides a well-defined framework for accessing the issues associated with software quality.

V. R. Basili, [23] proposed a model known as GQM (Goal Question Metrics)paradigm, which isastrongtoolforsoftwaremodelingandmeasurement.GQMoffersathoroughframeworkfordefiningandassessingsoftwareobjectives,questions,andmeasurements.Thisparadigmenablessystematicanalysisandimprovementofsoftwareprocesses,products,andservices. Softwareengineersmay use the GQM paradigm to minimize the complexity ofsoftware development while increasing the efficacy oftheir software modelsand metrics.

Robert B. Grady and Deborah L. Caswell,[24] viewstheenhancement of software development process in termsof quality, cost, and time necessitates the first steps ofissuediagnosis,establishmentofperformanceobjectives, and evaluation of software project efficacy.Organizationsmayenhancetheirlevelofsuccessbydedicatingresourcestowardstheestablishmentandexecution of a comprehensive software metrics program,which enables them to get a deeper comprehension oftheir software developmentprocedures.

M. Ortega et al. [25], designed a model prototype that reflects the essential attributes of quality. This model was evaluated using a method so it can be validated and also enhanced. The evaluation method consisted of: designing a survey, formulating, validating and applying the measurement instruments. When evaluating a software product using the model prototype, it was possible to ascertain its compliance with the standards and use the results to improve it. Since the evaluation was systemic, processes that affect certain characteristics of the product could be identified. Companies can benefit from the model proposed because it serves as a benchmark.

Reliability and **usability** are two critical attributes with all other attributes, because these attributes are directly impact the user experience, operational efficiency and overall success of software products. Ensuring high reliability and usability should be a primary goal in any software development company to meet user expectation, achieve business objectives and maintain a competitive edge in the market. Thus reliability and usability are two important attributes that contribute to the overall software quality.

Reliability refers to the ability of the software to perform its intended functions without errors. Usability, on the other hand, is concerned with how user-friendly the software is and how easy it is to use. Both reliability and usability are important for ensuring that the software meets the needs of its users and provides the desired level of quality.

Some of the research papers related to reliability and usability are discussed below:

J. R. Brown and M. Lipow,[26] proposes a model for evaluating reliability assessment and prediction with the help of development and validation. This paper analyses various paths through path testing as per flow diagram of none to node branching and evaluating with the help of probability distribution. It is important to evaluate the reliability of software products before their release to the public. Testing should also be performed at regular intervals to ensure that any software changes do not introduce new problems related with reliability.

Ronan Fitzpatrick and Catherine Higgins, [27] suggest that now it is time to review usability aspect and in order to measure usability which demands optimal attributes that must be measured along with required matrices to be expected. The aim of this paper is to identify all those attributes which should be measured. This paper also focuses on one of the important question “how the study of software usability has advanced over past twenty years by reviewing formal definitions of usability”? In order to ensure that software is both usable and of high quality, it is necessary to apply both European Community Law and Human-Computer Interaction principles and practices. This will ensure that software is designed and developed with expectation of end-user and provides a safe, efficient, and enjoyable experience.

ISO/IEC 9241-11 quality standard [28] defines a framework for understanding the idea of usability and applying it to scenarios, where consumers access interactive software systems. This standard includes three measures for usability: efficiency, effectiveness and satisfaction.

J. Nielsen, [29], explains iterative user-interface design which constitutes a pivotal component in the creation of successful user experiences. Through the adoption of a user-focussed approach and the employment of an iterative design process to refine and enhance the design, user-interface designers can effectively craft solutions that are customized to the specific needs, goals, and preferences of users. By understanding user needs and integrating them into the design process, user-interface designers can construct more efficient and intuitive solutions, thereby facilitating users' achievement of their objectives.

Nigel Bevan, [30], explain that the creation of a product or service permits consideration of various factors. Proper utilization of usability can guarantee ease of use and customization of the product or service to meet customer needs. It also facilitates universal access for all users regardless of their technical proficiency or ability. Optimal usability yields highest level of customer satisfaction, increased customer loyalty, and amplified profitability.

Manuel F. Bertoa and Antonio Vallecillo,[31] illustrate the significance of usability metrics in relation to software components. These metrics offer a means to measure the efficiency of a user interface and facilitate the identification of areas that require enhancement. Furthermore, they are an necessary tool for providing feedback to software developers and designers. As a means of ensuring an optimal user experience, usability metrics need to be incorporated into an organization's software development process.

Observation derived based upon the reviews of above research papers:

A software quality model is an essential component in guaranteeing software quality through the provision of an evaluative framework for enhancing software applications' quality. This model typically enclosed a range of quality aspects, metrics and measurement methods that enable organizations to assess their software's quality. The model serves to identify specific areas that require improvement, establish quality targets, and track progress towards achieving those objectives. Additionally, it facilitates organizations in comparing their software's quality with industry standards and best practices.

Quality factors may be treated as asset, if they are carefully designed according to domain of software application. The visualization of quality factors in a software application is depends upon the domain in which it is being utilized. For example in the healthcare applications, the quality factors included as accuracy, security, performance, user-friendliness etc.

The methodology utilized for metrics guided by software quality factors is conventionally grounded on universally-recognized benchmarks such as ISO/IEC-9126 standard. This includes the identification of the distinct quality factors appropriate to the software application, then design metrics that appraise those factors. These metrics methodology enclosed the execution of testing and analysis to accumulate data and authenticate the metrics.

2.3 APPLICATION OF SOFT-COMPUTING TO QUANTIFY SOFTWARE QUALITY FACTORS

Evaluation is the key aspect, which requires a proper mathematical tool. This mathematical tool will be responsible to convert qualitative aspects to quantitative assessment. Qualitative aspects to quantitative assessment require a suitable framework which accepts the input, process it and evaluates in the form of overall output.

The following research papers include various soft-computing methods for evaluation:
T.J. Ross, [32], explain that fuzzy logic has emerged as a strong tool for engineering applications due to its power in handling imprecise, incomplete, and uncertain information, rendering it suitability for several complicated real-world applications. Additionally, it can simplify certain control systems and offer well made solutions in the face of dynamic conditions. Its pliability, intelligibility, and potential to integrate with other technologies make it a compelling approach for a multitude of application domains. The progressive evolution of the domain of fuzzy logic is bound to discover novel prospects for engineering applications.

ReenaDadhich and BhaveshMathur, [33], concludes that the application of fuzzy logic can effectively measures the dependability of aspect-oriented software. It is observed by the authors that the utilization of fuzzy logic can present a more precise and dependable measurement as compared to the conventional techniques of software reliability evaluation. Moreover, the authors propose that this approach can be implemented for diverse software engineering activities, including testing, debugging, and maintenance. The authors further conclude that software developers can benefit from the use of this approach to increase the reliability of software.

K. K. Aggarwal et al. [34], proposes four factors which affect maintainability. This paper explains about four factors LV, LS, ACC, CR which acts as an input for the proposed fuzzy model. This paper uses centre of gravity method for defuzzification to quantify maintainability. This paper quantifies maintainability with the help of fuzzy logic and results prove that the measurement obtained from this model has a strong co-relation with maintenance time.

James Senior et al.[35], proposed a method which analyzed and visualized overall metric score so that project managers or stakeholders can easily evaluate the expected performance. The solution proposed here visualized in the form of tree structure with the help of fuzzy logic. Fuzzy logic was employed in the calculation of metric tree scores. The fuzzy inference engine used to meet the needs of the customer. Fuzzy logic evaluated in metric analysis, Global Tech demonstrates its efficiency and potential.

Harish K. Mittalet. al [36], proposed a way to quantify quality of software modules on the basis of inspection rate and error density of the software product with the help of fuzzy logic. This paper uses triangular fuzzy function to quantify two metrics inspection rate and error density/KLOC of software product. Software modules evaluated with the help of quality grades by using fuzzy logic. Fuzzy logic has its own advantage to convert qualitative aspect to quantitative evaluation with the help of fuzzified inference rule. Software modules are rated on scale of 1 to 10, where scale 1 is supposed to more error prone. The paper states that proposed model is generalised in nature hence it is used in other area of software engineering.

Deepak Guptaet.al [37], presents a case study of different software quality estimation technique to propose a framework of different quality models and compare the performance of each one, some of them are ANN, Case-Base rule, rule based system, regression tree, multiple linear regression and fuzzy system. This paper states that rule based system technique and fuzzy technique are two most prominent techniques for designing a software quality model. This paper concludes that single technique is not effective as compare with combination of techniques such as fuzzy and rule based technique are more effective than fuzzy technique or rule based techniques individually.

SaumendraPattnaik et al. [38], explain that the use of fuzzy logic is required for any kind of quantitative or qualitative analysis. Since the evaluation process contains ambiguity, fuzzy logic is one of the greatest techniques for dealing with such situations.

T. L. Saaty [39], proposed an evaluation procedure, which is determined by the relative importance of the many quality parameters. The first step toward a more effective assessment procedure will be determining the ideal value of the weights.

AHP, which stands for "analytic hierarchy process," is one of the ways in which pair-wise comparison is conducted to determine the relative significance or preference of items by comparing them in pairs. This is accomplished by giving numerical numbers that signify the intensity of preference. These values are often represented on a scale, such as the Saaty scale, where a value of 1 implies equal significance, but other values imply different levels of preference. This approach helps where anyone may determine an acceptable weight by doing pair-by-pair comparisons.

J.J. Buckley [40], extends the concept of the conventional AHP methodology by using fuzzy logic, known as fuzzy analytic hierarchy process (FAHP). Fuzzy logic enables the incorporation of ambiguity and inexactness in the process of decision-making. In a fuzzy Analytic Hierarchy Process (FAHP), the pair-wise comparison matrix is expanded to include fuzzy numbers. Instead of giving an exact value, decision-makers have the option to articulate their judgments using language phrases such as "slightly more important" or "strongly more important," which are then translated into fuzzy numbers.

Observation derived based upon the reviews of above research papers:

Most of the researchers found that fuzzy logic is the most well known soft–computing method. Fuzzy logic is one of the mathematical approaches which have capability to convert qualitative aspectsto quantitative assessment. The literature survey found that fuzzy logic is one of the prominent mathematical approaches to deals with vague and ambiguous situations of real world problem.

2.4 APPLICATION OF MCDA (MULTI CRITERIA DECISION ANALYSIS) FOR QUANTIFICATION

A real world problem depends upon many criteria's, where these criteria's are conflicting to each other. This kind of problem always is solved by MCDA, which proposes a proper decision making keeping in view about all conflicting criteria's. Fuzzy MCDA is one of the effective tools which incorporate MCDA along with the impact of fuzzy logic.

ChristerCarlsson and Robert Fuller [41], inferred that fuzzy multiple criteria decision making represents a potent and promising tool for tackling decision making in intricate problem domains. This approach has the ability to furnish decision makers with a comprehensive framework for addressing the intrinsic uncertainty and vagueness of the decision-making scenario. Additionally, recent advancements in this domain have bolstered the potential of this technique in providing efficient solutions to multi-criteria decision making challenges.

Salvatore Greco et al.[42], proposes the set of decision rule for MCDA. This paper uses rough set theory to design MCDA. Two important properties uses by paper:

1. Calculation with the help of dominance relation which deals with preference order properties of criteria.
2. Analyze pair wise comparison table, allow handling preference relation for choice and ranking.

The decision rule approached by this paper in the form of "if then ". This proposed model used in the paper based on elementary concepts and mathematical tools.
With the increasing complexity of decision making problems, the use of these sets can be invaluable for decision makers.

Michel Grabisch and Marc Roubens[43], proposes Choquet integral as a general tool to deal with multiple criteria decision making. This paper discussed theoretical aspect of choquet integral and then discussed practical problems in the form of fuzzy measures. This paper proposes two approaches MSE method and CS method, where MSE method always gives a solution, which identifies a hidden decision behaviour. The CS method based on ordinal information. The advantage of CS method over MSE method is that, it does not violate the ranking provided by decision maker. This method is more suitable than MSE.

G.Buyukozkan et al.[44], proposed a selection framework based on fuzzy multi criteria evaluation approach to improve quality of decision making in the development of software project. This model based on fuzzy AHP to access tangible and intangible qualitative factors. The combine effect of two factors will give alternatives for software development strategy to select the most appropriate among existing. The effect of the model will be demonstrated with real time case study. This paper concludes that the ambiguities involved in the assessment effectively represent, which increases the efficiency of proposed framework, thus produces effective decision making.

Ajit P. Singh and A. K. Vidyarthi [45], conclue that the Fuzzy Multi Criteria Decision Making (FMCDM) approach is a viable approach for the optimal allocation of landfill disposal sites. FMCDM combines the advantages of both the crisp multi-criteria decision-making (MCDM) approach and the fuzzy logic approach, allowing for the consideration of fuzzy criteria and factors. This approach is effective in taking into account both technical and non-technical factors and provides an effective decision-making framework for landfill site allocation.

P. R.Srivastava et al.[46], proposes fuzzy multi-criteria decision deals with the uncertainty due to qualitative perception and intuition of decision makers involved in the process of development of software product. We know that development of software product depends upon certain time-frame and budget; with-in that error free software must be delivered, so that paper focused to evaluate preference degree of each alternative and to select most appropriate among all alternatives. This paper also states that mathematical concept of MCDM as well as proposed method applicable to real world problem where values can be either numerical or linguistic. This paper concludes that this work can be extended by evaluating the sub-characteristics of software quality model.

Sanjay K. Dubey and Disha Sharma,[47], proposed an evaluating model, with the help of ISO/IEC-9126 where ISO/IEC-9126 acts as baseline model. The proposed model design with the help of ISO/IEC-9126 and 3 new sub-characteristics flexibility, robustness and supportability. The proposed evaluating model evaluate with the help of fuzzy multi criteria approach. The proposed model has better performance than the ISO/IEC-9126 model. Performance will be compared by evaluating the crisp value of net software quality for both ISO/IEC-9126 model and proposed model, with the help of defuzzification. The results of this research provide a useful platform for further study on software quality appraisal.

FarhadLotfi et al.[48], proposed a way to determine fuzzy weights, with a specific emphasis placed on the fuzzy least square error approach and the fuzzy BWM (Best Worst approach). The purpose of this chapter is to provide readers with a better knowledge of fuzzy weight determination techniques and the possible applications of these approaches in decision-making situations that occur in the real world.

P.R.srivastava and K. Kumar [49], explain about three perspectives, a project manager, a developer, and a tester, and further developed the research. However, the focus of these studies remains on developing a framework to ensure the production of high-quality software products and quantifying them as desktop applications.

P. R.Srivastava et al. [50],proposes quantification of qualityfactors. The quantification of quality factors investigates a method to examine quality variables or justify overall quality while the software product is being developed. This paper examines the application of a fuzzy multi-criteria approach to quantify software quality.

Observation based upon the reviews of above research papers:

Researchers found that real-world problem naturally support multiple criteria. Thus multi-criteria assessment is a decision making process used to evaluate and prioritize it. Multi-criteria assessment can be implemented by Fuzzy AHP, Fuzzy TOPSIS etc.

2.5 QUANTIFICATION OF SOFTWARE QUALITY FACTORS FOR MOBILE APPLICATION

Quantification of software quality factors is necessary to understand the quality expectation of mobile application during its development by considering its limitations such as screen size, memory, mobility etc. A framework must be designed which taking care of such limitations. Quantification of software quality factors provides a structured approach to measuring, analyzing, and improving the various aspects of quality of mobile application. Quantification is a systematic approach which, improve and maintain high-quality standards in mobile applications.

AzhamHussain and Maria Kutar [51], proposed a framework for determining the usability of mobile phone applications is a powerful instrument for evaluating applications' usability. The framework offers an all-encompassing array of measurements that gauge mobile phone applications' usability, presenting developers and testers with a structure that guarantees usability and fulfills user demands. The suggested framework can enhance the user's application experience, decrease expenses related to the development process, and secure that applications fulfill user requirements.

Euler HortaMarinho and Rodolfo Ferreira Resende [52], assert that optimal approaches for the development of mobile applications necessitate the incorporation of quality factors to guarantee that mobile applications fulfill the demands of users and offer the utmost level of gratification. Quality factors, such as usability, performance, security, maintainability and scalability, ought to be taken into account. Furthermore, it is indispensable to integrate quality assurance and testing into the development process so as to ensure that mobile applications are of the highest quality and provide an optimal user experience.

Ali Idriet. al[53], proposed the utilization of ISO/IEC 9126 as a benchmark for software quality in mobile environments is a feasible and efficient approach to guarantee the quality of mobile software products. The standard is all-encompassing, and its implementation has the potential to enhance the usability, dependability, efficiency, and maintainability of mobile software products. This paper suggests that the adoption of the standard can result in elevated customer satisfaction and the deployment of mobile software products.

Maximilian Wich and Tommi Kramer[54], states about the usability aspect of mobile business applications. The survey instrument, developed as part of this study, has been established as a legitimate and dependable means to assess the usability of mobile business applications. The instrument is exhaustive and adaptable, making it suitable for evaluating the user experience of mobile business applications in diverse settings. Furthermore, the instrument can aid the design process of mobile business applications by identifying areas requiring improvement. The implications of this study's results are anticipated to be useful in the creation of effective mobile business applications in the future.

S. Panichella et al. [55], explain that user reviews can serve as a valuable repository of data for software maintenance and evolution endeavors. Through the utilization of Machine Learning methodologies, it is possible to effectively categorize user reviews into distinct classifications pertaining to software maintenance and evolution. With continued refinement and enhancement, this framework could be implemented across other software platforms, thereby aiding developers and software maintainers in upgrading their products.

ZuhairElkheir and Ariffin Abdul Mutalib[56], observed that mobile learning applications present a considerable potential for improving learning and collaboration in diverse educational settings. However, the development of such applications is a demanding task that necessitates consideration of various aspects, including user experience, privacy and security, assessment, and collaboration. This investigation has furnished a comprehensive overview of the contemporary research on mobile learning applications and the challenges linked with their design. It has also indicated the requirement for further research aimed at advancing the efficacy of mobile learning applications and addressing the assorted challenges and design considerations.

André Nitze and Andreas Schmietendorf [57], explains that the perceptions and anticipations of mobile users with regards to software quality exhibit notable variations, contingent upon diverse factors such as the platform, age of the user, and the type of application employed. The findings of this study are instrumental in guiding mobile application developers to enhance their ability to meet the quality expectations of end-users and augment their overall user experience. Moreover, these results furnish valuable discernment into how mobile users assess and construe software quality, thus enabling developers to devise applications that cater to the preferences and necessities of users.

PauliusPaskevicius and RobertasDamasevicius [58], explain that the creation of a functional interface for a mobile e-commerce system is a multifaceted endeavor that necessitates the consideration of numerous factors, including but not limited to usability, accessibility, and security. By taking into account the needs of the end-user and incorporating the most current mobile technologies, designers can develop an interface that is both user-friendly and enjoyable. With the appropriate methodology, the interface will be able to provide the end-user with an experience that is secure, efficient, and pleasurable. Through comprehending the demands of the end-user and constructing a functional interface, developers can establish an e-commerce system for mobile devices that is successful in providing a superior user experience.

Lawrence Barnett et al.[59], has demonstrated that the efficacy of mobile games is significantly influenced by user-friendly navigation, clear instructions, and appropriate feedback, all of which are usability factors. Additionally, the quality of graphics and overall game design are crucial elements that significantly impact the user's experience. Based on the results of this study, mobile game developers can enhance their game design and thereby create a more engaging user experience. Furthermore, further research is necessary to comprehensively comprehend the user experience of mobile games.

A. Hussain et al. [60], proposes an assessment of the e-reader application predicated on the mobile experience. Several features, such as page-turning, bookmarks, and collections, enhance the enjoyment of consuming digital literature. Moreover, the application boasts a user-friendly interface that is facile to navigate, with an uncomplicated and uncluttered design. To encapsulate, this application provides a remarkable reading experience that is sure to be savored by e-reader enthusiasts.

S. Shafiq and T. A. Khan, [61], emphasized the significance of usability in mobile educational applications. Specifically, this paper investigates the usability of chosen mobile educational applications in the domains of mathematics and reading skills, with the objective of detecting the usability issues identified by the participants. Subsequent to usability assessment through the testing of these mobile applications on school children, comparisons are drawn between the results obtained from usability and learning evaluation to substantiate our hypothetical claim that usability has a substantial effect on learning.

David Parsons and HokyoungRyu [62], endeavors to explore the means by which quality can be evaluated within the framework of mobile learning design. Our analysis considers the specific areas within our framework that may be assessed using established quality metrics. Additionally, we offer a discussion on the further design considerations that must be taken into account during the M-learning system development process, including the user's role and collaborative support, which have not received the same level of attention as the technological aspects of the M-learning context. Lastly, we introduce three metrics that we believe could supplement the current ISO/IEC 9126 metrics for evaluating M-learning environments: metaphor, interactivity, and learning content.

KarimaMoumane et al.[63], presents an empirical study with the help of a framework which measures and evaluates usability expectation of two mobile applications, Google apps and Google maps running on different mobile OS, such as android, ios and Symbian. The framework designed according to ISO/IEC 9126 quality standard. This paper evaluates usability expectation with feedback of 32 users and explores usability issues related with hardware and software. The feedback will be taken care by designers and developers to take corrective measures.

Fernando Rosell-Aguilar[64], presents an evaluation framework which evaluates language learning app on the basis of four categories i.e. technology, pedagogy, user experience and language learning along with set of criteria within the categories.

SerenBasaran and Firass El Homsi,[65] presents a decision making mathematical approach Fuzzy TOPSIS to implement MCDM. This paper evaluates ranking within six mobile mathematical learning applications. This paper adopts ISO25010 software quality standard. The aim of this paper to retrieve the best alternative among present applications. This approach helps to decision makers. Mobile game is one of the fast growing businesses.

Paula Bitrián et al.[66],focused upon"how to engage more and more users through gamification?". This paper conduct an analysis with the help of data from 276 users of mobile gamified app and conclude that gamification depends upon three factors need for competence, autonomy and relatedness. This paper also provide a theoretical and practical implication, which helps to developers so that they think about more effective mobile game.

Manish Mishra and ReenaDadhich [67], presents a complete framework to convert qualitative aspects to quantitative assessment and evaluates over-all quality for mobile application with the help of Fuzzy logic.

Manish Mishra and ReenaDadhich [68], presents a complete framework to convert qualitative framework to quantitative measurements, which design a framework for usability expectation of M-Commerce application with the help of GQM and ISO/IEC-9241-11 along with the assistance of fuzzy logic.

Manish Mishra andReenaDadhich[69], presents a novel fuzzy based mathematical algorithmwhich,evaluates relative order in the software quality factors for usability expectation of mobile game with the help of pair wise comparison and with this information, decision maker is able to allocate appropriate fuzzy weight.

HaiqiFenget al. [70], suggest that Firmsmustunderstandcustomerneeds,dothoroughmarketresearch,andstaycurrentontechnologytosucceedinthisfield.Businessesmustinvest in research and development, customer service,and marketing to succeedin mobilecommerce. Today, users may use their mobiledevices to make online purchases, pay bills, and managetheiraccounts.Mobilecommerceiscomplicatedanddependsonseveralfactors.

Mengtian Cui andLibo Zhu, [71] present a model for determining the most effective user interface from a variety of alternatives by analyzing user interviews and questionnaire responses and satisfying user expectations for a mobile game application. Mobile gaming is one of the most complex domains, where considerable effort was required to ensure that mobile game applications were usable.

PawelWeichbroth [72], suggests that, the ease of use of a mobile gaming application is and will continue to be a key factor, particularly for first-time users and other inexperienced players. In order to have a good usability quality framework, one must ensure that the optimal amount of quality elements must be present.

SorinNădăban et al. [73], presents a survey paper that will explore the decision making strategy with a real world problem which has multiple criteria. This paper presents a decision making for MCDA with th e help of fuzzy TOPSIS method. This paper offer a general view of the developments of fuzzy TOPSIS methods, literature review and explore different fuzzy models that have been applied to the decision making field along with some applications of fuzzy TOPSIS.

J.R.S.C. Mateo [74], presents two methods, weighted sum method and weighted product method used for evaluation in the problem of multi criteria analysis in renewable energy industry.

Observation based upon the reviews of above research papers:

Mobile application is one of the prominent areas which are expended exponentially in the recent years. Now mobile application has countless area of application. This popularity enhances the competition within the mobile applications market. Customers have so many choices for a mobile application within one particular domain. Thus it is required to observe overall quality of the mobile application under development. The quantification of quality factors ensures about the quality of mobile application with customer expectation.

2.6 RESEARCH GAP

This has been noted as limitations of the existing literature according to the literature review done for the purposed research work:

1. Quality frameworks for mobile applications or qualitative aspects have been discussed in an existing research study, but it lacks with complete quality framework according to the mobile application development company.

2. The quantitative assessment of desktop-based applications were the main focus of the majority of the study article, whereas mobile applications were considered less important.

3. The process of quantification is function of inputs: fuzzy rate and fuzzy weight. Nearly all of the papers analyze fuzzy weight, like fuzzy rate. The purpose of fuzzy weight will be entirely different than fuzzy rate, thus it is necessary to take into consideration fuzzy weight, while keeping in mind the relative importance among the quality factors.

4. The existing body of literature does not adequately address about the views of all entities of mobile development company involved with in the development of mobile application, despite the fact that the complete scenario of any mobile application development company is dependent on the perspectives of developers, testers, and end-users, as well as the perspectives supplied by stakeholders, investors, and quality managers who are referred to as decision makers.

The following points are considered for research gap, in view of above mentioned limitations as per literature review.

1. Complete generic framework for evaluating the quality of mobile applications that are customized to specific domain is not present.

2. Optimal set of quality metrics those are specific to the domain.

3. Quality metrics according to quality standards, which are centered on the developer, tester and β-users.

5. Designing of most significant metrics together with opinion of professionals such as Security Metrics.

7. To identify the limitations of the current methodology and develop a benchmark.

8. Evaluate the quality assessment based on the effects over the long term.

9. Assessment of Overall Quality based on several platforms (Operating Systems and Hardware).

10. Recommendations for the mobile application business and areas of further research.

2.7 SUMMARY

This chapter explores various observations of the researchers regarding the quality aspects of software as a product. These observations are divided in to four distinct categories. First category discuss about quality framework which has suitable number of quality factors along with their metrics. Metrics are basic building block which strengthen quality factors and thus converted in tosuitable overall quality framework. The second category discuss about the suitable soft computing technique which evaluates the quality framework and thus qualitative aspects converted in to quantitative assessment. Most of the researchers found that fuzzy logic is one of the appropriate soft computing techniques. Third category discusses the application of multi criteria decision approach for the purpose of quantification. Forth category discusses quantification of software quality factors for mobile application. Many researchers have their views about framework and quantification of quality factors for mobile application. Most of them focused on usability framework and evaluation. It is observed that no framework or methodology is available which assess the overall quality of mobile application being developed by considering all the entities i.e. investors, stakeholders, quality managers, developer, tester and β-users of a software development company.

CHAPTER 3

SOFTWARE QUALITY FACTORS FOR MOBILE APPLICATION

3.1 INTRODUCTION

The quality of the application is an essential factor to consider while evaluating software. Both software developers and users have a huge difficulty when it comes to the assurance of the quality of software. This is because there are a great number of software quality factors that come into play when determining the quality of software. Some of these factors are very important and they become the basic building block for overall quality of mobile application under development. A significant amount of the program's effectiveness, or lack thereof, is determined by these factors. As a consequence of this, it is of the utmost importance to develop strategies during software development process [24] that improve the quality of the program and cut down on mistakes. The software developers and designers of an organization do significant research and make use of a variety of techniques in order to assess and record the quality of their software products before they are released to the market [25]. As an additional point of interest, they do routine quality assurance checks on their software products across the whole of the software life cycle, looking for techniques and criteria that improve the overall level of quality. There is a widespread consensus that the quality of a program is contingent upon a number of different characteristics that may be used as metrics to evaluate the overall quality of mobile application programs that are in the process of development. The aspects that are included in these factors include those that are immediately observable, such as logical flaws, as well as those that are indirectly measurable termed as functional and non-functional aspects.

In order to access the quality of mobile application, only specific software quality factors are taken into account including essential metrics [20, 21]. In addition to functioning as a trustworthy indication of the demand for software development and maintenance, software quality factors are an essential and ideal measure for evaluating the quality of mobile application [52]. Within the corporate sectors, where businesses are expanding and making efforts to improve the quality of their operations, the relevance of software quality factors are growing, and they are also becoming more widely accepted. Quality measurements, on the other hand, are quantitative evaluations of the degree to which a program addresses a particular quality factor that has an effect on the program's overall quality.

This chapter divided in to 5 sections 3.1 to 3.5. Section 3.1 introduces the importance of software quality factors for any mobile application. Section 3.2 explains about significance of quality factor with in mobile application along with impact with some important quality factors. Section 3.3 discuss about designing of quality factors, while section 3.4 discuss about designing of software metrics for mobile application. Provide an explanation based on a case study which establishes the appropriate qualitative usability framework for mobile game application along with relevant metrics.Section 3.5 concludes with summary of this chapter.

3.2 IMPACT OF QUALITY FACTORS IN MOBILE APPLICATION

When it comes to determine the overall quality of any software application, quality aspects are always going to play a significant role. The quality of software is a multifaceted notion that enclosed a variety of attributes as quality aspects, and the ultimate objective is to bring all of these quality aspects together to generate a product of high quality. The aspects of quality contribute to the identification, access, and assurance of the standards that are required. Different generic software quality standards, each define their own unique collection of quality parameters. There are a great number of these standards [1, 15, 19]. Considering that these quality parameters are applied according to domain, it is inevitable that they will always come up with customization. The purpose of this research work is to examine one of the most prominent quality standards, which is used by industry as well as in theoretical aspects like ISO/IEC-9126 standard [15]. A number of quality factors that contribute to an application's overall performance, user experience, and market acceptability all plays a great role in determining the overall quality of mobile application according to customer expectations and theirrequirements.

The following is a list of some of the important quality factors and their impact on development of mobile applications, if the quality factors will not be considered adequately:

1. Performance

Impact: It is possible for users to get frustrated and quit a product if the performance is slow or inefficient.

Things to Take into Account: Response time, load time, and general responsiveness are all very important considerations with regard to performance. One should undertake performance testing, optimize the code, and reduce the amount of resources needed.

2. Reliability

Impact: Unreliable applications may lead to unpredictable results,loss of data, and thus may lead to bad impression.

Things to Take into Account: It is important to take into consideration that doing exhaustive testing, addressing errors, and performing frequent updates to repair problems all contribute to enhanced reliability.

3. Usability and User Experience

Impact: User-friendly Interfacesincreases levels of customer satisfaction and retention.

Things to Take into Account: Design, navigation, accessibility, and feedback systems are fundamental components that must be taken into consideration in order to provide a user satisfaction.

4. Safety and Protection

Impact: Having security flaws may result in data breaches, identity theft, and harm to the app's reputation, among other negative consequences.

Things to Take into Account: It is important to take into consideration the following: implement safe coding methods, make use of encryption, conduct frequent security audits, and ensure that software dependencies are kept up to date.

5. Scalability

Impact: If the existing network architecture is not able to manage a rise in the number of users or the amount of data, then this may lead downtime or a decline in performance.

Things to Take into Account: The architecture of the application should be designed in such a way that it supports SOA, distributed network, load balancing, scalable database, auto scaling, CDN, message queue, caching etc., which makes a system scalable and reliable.

6. Compatibility

Impact: If the application is not compatible with a variety of different hardware configurations, operating systems, or screen sizes, it may restrict its accessibility.

Things to Take into Account: One thing to keep in mind is to test the application on a variety of devices with different hardware configuration, operating system versions, and screen sizes. Maintain compatibility with well-known frameworks and libraries that are provided by third parties.

7. Maintenance

Impact: There is a possibility that the app may become obsolete if it is difficult to maintain and update it.

Things to Take into Account: It is important to make use of code that is both modular and well-documented, to adhere to the best practices for coding, and to make preparations for future upgrades and improvements.

8. Compliance

Impact: In any case when legal and regulatory criteria are not met, it may result in legal complications and negatively impact the reputation of the application.

Things to Take into Account: Keep updated on the laws and regulations that are relevant as per current scenario, and put into place the required compliance procedures, such as privacy and data protection requirements.

9. Testability

Impact: Inadequate testing might lead to faults and problems that are not found right away.

Things to Take into Account: Important considerations include the creation of exhaustive test cases, the implementation of automated testing where it is feasible, and the execution of exhaustive testing across the whole SDLC process.

10. Efficiency in Cost Management

Impact: Apps that are not properly optimized may result in increased hosting costs and a decrease in profitability.

Things to Take into Account: There are a few things to take into consideration, including optimizing resource use, selection of hosting options that are cost-effective, and thinking about the long-term financial ramifications of design choices.

When it comes to development of mobile application that is both profitable and sustainable, it is very necessary to address these quality factors across the whole SDLC process. In order to maintain and improve the quality of amobile application over time, it is essential to provide regular updates and to be sensitive to the feedback provided by developer, tester and β-users.

3.3 DESIGN OF QUALITY FACTORS

The creation of software includes a qualitative aspects known as the design of quality factors [14]. In addition to the functional requirements, the roles have much more importance for non-functional requirements [29]. These quality factors must be satisfied in order to create a high quality software product. It is necessary to maximize these qualitatively defined quality factors and meticulously design a framework that is readily measurable in order to guarantee that all of the human resources connected with software development process are able to produce high-quality software.

The design of quality factors is a continuous and ever-changing process that requires the involvement of all stakeholders, including developers, testers, and β-users, in addition to quality assurance team. Therefore, it is essential to establishcohesion among all of them in accordance with the objectives and limitations of mobile application development. The whole of this effort will always be appreciated when the developed application eventually provides a user friendly interface and usability expectation of its users as well as the requirements of all stakeholders.

3.4 DESIGN OF SOFTWARE METRICS FOR A MOBILE APPLICATION

Software metrics are essential for the foundation of any mobile application. Designing this part requires careful consideration of software metrics [17], as they represent important quality factors. The following case study demonstrates the process of designing metrics for a mobile game to meet usability expectations.

3.4.1 Design of software metrics for assessing the usability of mobile game: Case Study

Usability refers to the ease of use and overall user experience of a software product. It examines how effectively a customer can utilize the interface to execute activities efficiently, happily, and without irritation? [30, 51]. Usability metrics are utilized to assess how simple and effective the software product for customers. These metrics provide an approach to measure the effectiveness of usability [31, 54].The usability assessment of a mobile game has a significant influence [59, 71, 72].

This section introduces a fuzzy-based assessment framework for assessing the expected usability of mobile games [66]. In order to do this, anISO/IEC 9126 [15]have been considered as a quality standard. According to ISO/IEC 9126 [15], there are four quality factors defined for usability that are to be extracted from thisquality standard. These quality factors are learn-ability, understandability, attractiveness, and operability. There are seven other quality factors that are need to be taken into consideration: simplicity, memorability, efficiency, attractiveness, challenge, user happiness, and error and recovery to complete the usability assessment framework as shown in Table 3.1. There are two fuzzy inputs: fuzzy weight and fuzzy rate, for fuzzy weight it is necessary to make a decision based on priorities of quality factor, and the fuzzy rate is determined through the questionnaire responses filled or submitted by β-users. This case study takes into account the experiences of both novice and expert users as shown in Table 3.2. Five scale triangular fuzzy functions, VH (very high) to VL (very low) are used in order to consider for both type of input parameters.

Table 3.1: Usability Attribute Along with Quality Factors

Attribute	Quality Factors
Usability	Learn ability
	Understand ability
	Simplicity
	Memo ability
	Efficiency
	Attractiveness
	Challenge
	User satisfaction
	Effectiveness
	Error and recovery
	Operability

Table 3.2: Quality Factor according to users

SN	Quality Factors	Type of user
1	Learnability	Novice user
2	Understandability	Expert user
3	Simplicity	Novice user
4	Memorability	Expert user
5	Efficiency	Expert user

6	Attractiveness	Novice user
7	Challenge	Expert user
8	User-satisfaction	Expert user
9	Effectiveness	Expert user
10	Error & recovery	Expert user
11	Operability	Novice user

The remaining part of this section gives a short description of each and every quality factor given as:

1**. Learnability:** Learnability defined as the degree of easinessto learn a mobile game without any help. It is animportant quality factor for the novice user, Novice users always search for easy to learn model. Although differentalternatives are available in any mobile game in the form of informative text, demo etc. to understand the game but anovice user always go for natural learning as per user's intuition. Metrics are designed for mobile game as shown in Table 3.3:

Table 3.3:Metrics of Quality Factor Learnability with its Description

Metrics	Description
Numberofattempts	Gamehasoptimalnumberofattemptstofinisheach task of each level.
Probabilityforerrors	Probabilityforerrormust be in lower side.
Duration for searchingabutton	Timemustbeminimum for searchingabuttontoperform specific function.
Sufficientassistance	Inbuiltassistmustbeprovidedwhichhelpsusers specially novice, during execution of game.

2. Understandability: Understandabilitydefinedasunderstanding of logicalconceptsforeachtask.Itisanimportantquality factors for the expert user. Metrics are designed for mobile game application as shown in Table 3.4:

Table 3.4: Metrics of Quality Factor Understandability with its Description

Metrics	Description
Identification of pattern	Recognizethepatternwhichappearsamongalltask or level
Self descriptive icons	An icon of each game player represents "what they do in the game"?
Reaction of environment	Reactionofenvironmentalobjectfollowdescribedgame rule as per user action.
Skillsetof game player	Properdescriptionofskillsetof game players intheformof textual information, demonstration etc.
AI logic	AI is included or not in the mobile game
Flow and logic	Expert user easily understandsflowandlogicofmobilegameinminimum time.
Sufficientknowledge	Properdescriptionofmobilegameintheformoftextual information, demonstration etc.
Rules to score	Properdescriptionhowtoscorepoints within the mobile game

3. Simplicity: Simplicity defined degree of being easy to understand design. It is an important quality factor for the novice user. Metrics are designed for mobile game application as shown in Table 3.5:

Table 3.5: Metrics of Quality Factor Simplicity with its Description

Metrics	Description
Description of options	Menu options are descriptive in such a way that user can easily understood.
Characters are interactive	Characters attracts novice user
Navigation for destination	Proper navigation to reach destination
Modeling of symbols	Symbolicrepresentationwillbesameforeachtask or level
Compatible with screensize	Mobilegameinterfacecompatibleforeachscreen size
Proper orientation	Mobilegameinterfaceproperlyalignedasportrait or landscape.
Compatibility of key board	Key board will be compatible as per the size of the screen.
Time to search particular button	Avg.timewillbeminimizedtosearchparticularbutton for a first time user
Zoom in or out	Itishelpfulandnecessarytovisualizeamapbased mobile game.

4. Memorability: is defined as the degree of easy with which a user efficiently can remember how to use an application.Itis basically trade off in between time and recall pattern. This attribute specially design for expert user. Metrics aredesigned for mobile game application as shown in Table 3.6:

Table 3.6: Metrics of Quality Factor Memorability with its Description

Metrics	Description
Recallpattern for hours	Doesexperienceduserrecallallfunctionsassociatedwithataskwhichwasinterruptedby him for few hours?
Recallpatternfordays	Doesexperienceduserrecallallfunctionsassociatedwithataskwhichwasinterruptedby him for few days?

5. Efficiency: Efficiencyis defined as ability of user to complete a task with speed and accuracy. This quality factor specially designs for expert user. Metrics are designed for mobile game application as shown in Table 3.7:

Table 3.7: Metrics of Quality Factor Efficiency with its Description

Metrics	Description
Duration onscreen	Timedurationmustbeoptimizedasperthenature of mobile game for each level.
Durationtocompletetask	Timedurationmustbeoptimizedasperthenature of mobile game for all task of each level.
Average number of errors	Average number of errors must be minimized.

6. Attractiveness: Attractiveness is defined as degree to which application has been made attractive. This quality factor speciallydesignsfornovice user.It is tradeoffinbetweentotaltimespentandrelaxation.Metricsaredesignedformobile game application as shown in Table 3.8:

Table 3.8: Metrics of Quality Factor Attractiveness with its Description

Metrics	Description
Eventbasedsoundeffect	Auniquesoundappearswheneveraneventoccurs.
Font of text and graphics	Font of text chosen as per the theme of game,whilegraphicssuitableforlatestgraphicshardware.
Nightmode	Night mode available to play the game in the night.

7. Challangeability: Challangeabilityis defined as to maintain the challenge for every one level increment i.e. from n^th^level to n+1^th^ level. This quality factor specially designs for expert user. Metricsaredesignedformobile game application as shown in Table 3.9:

Table 3.9: Metrics of Quality Factor Challangeability with its Description

Metrics	Description
Degreeofchallangebility	Challangebilitymaintainedforeachlevelofincrement.
Consistencyinprogressivelevel	Difficulty maintained as per level.

8. Satisfaction: Satisfaction defined as per user's perceived level of expectation and need. Metricsaredesignedfor mobile game application as shown in Table 3.10:

Table 3.10: Metrics of Quality Factor Satisfaction with its Description

Metrics	Description
Execution of allfunctions	IfansisyesthenfuzzyvalueisVHotherwisefuzzyvalue is VL.
Experience asperexpectation	Overall experience of the game as per expectation regarding its peers.
Gaming experience	User satisfied with over all experience with the game.
Overall user interface	User interface will be comfortable regarding long duration of the game.
Overall functionality	Overall perception of playing the game.
Overall challenge	Overall Challenge experience in the whole game

9.

Effectiveness:E ffectivenessisdefinedasabilityofusertocompleteataskingivencontext.T hisquality factorspecially designs for expert user. Metricsaredesignedformobile game application as shown in Table 3.11:

Table 3.11: Metrics of Quality Factor Effectiveness with its Description

Metrics	Description
Numberofsuccessfullycompletedtas k	Iseachandeverytaskwithitsspecificcontextexec uted as per the requirements?
Number of steps	Numbers of steps are optimizedasperthecontext for a particular task.

10. Error and Recovery: Error defined asamount and type of errors whichoccur during task execution. Metricsaredesignedfor mobile game application as shown in Table 3.12:

Table 3.12: Metrics of Quality Factor Error and Recovery with its Description

Metrics	Description
Recoverfromerror	Definedasrecoverabletaskdividedbytotalnoof errors
Errorrelatedwithdeveloper	If it is yes then very serious issue
Warningmessages	Warningmessagesproperlyunderstoodbyfirsttime user
Interrupts during game	Offlinegamepausedandrestartduring Incoming voice call or videocall, if yes then fuzzy value is VH, otherwise VL

11.

Operability: Operabilitydefinedasrelativeeaseoflearningapp.Metricsaredesignedformobile game application as shown in Table 3.13:

Table 3.13: Metrics of Quality Factor Operability with its Description

Metrics	Description
Optionforcolorblindperson	Optionavailablesothatacolorblindperson plays the mobile game.
Action& reaction	Eachandeveryactionproperlyrespondsasperthe rule.
Response	Proper response in terms oftime and reactionwhenever user operates with buttons (softwareor hardware)
Web browser version	Web browser version is available , if yes then fuzzy value will be VH else VL
Compatible for OS	Compatible for each mobile operating system, If yes then fuzzy value if VH else VL

The case study mentioned above is being explored here, where metrics will be developed according to the mobile application that is particular to the domain. In order to facilitate the process of converting qualitative aspects into quantitative assessment, these metrics have been created in such a manner that they can be easily transformed into a fuzzy-based mathematical framework. This framework will be used for computation and evaluate numeric value, both of which are components of the process.

3.5 SUMMARY

This chapter explains the importance of software quality factors along with metrics. The frameworks for designing the metrics are dependent upon particular quality factor and thus carefully designed as per the current scenarios. Today’s market is a highly competitive environment. This is taken into account as an important approach in the development of metrics. This chapter also designs a usability framework for mobile game application as a case study, which will be further converted form qualitative aspect to quantitative assessmentand hence compute crisp value which will be used for quality enhancements.

CHAPTER 4

FUZZY LOGIC BASED QUANTIFICATION FOR MOBILE APPLICATION

4.1 INTRODUCTION

This chapter presents one of the prominent soft computing approach fuzzy logic, to quantify software quality factors for mobile application [37]. Fuzzy logic is a powerful approach for problem-solving that may be used in variety of fields, including embedded control and information processing, amongst others [32]. When compared to traditional logic, fuzzy logic provides a more user-friendly approach to the process of drawing definitive conclusions from data that is very imprecise, unclear, and confusing. Fuzzy logic helps to ease the analysis of imprecise data in order to arrive at exact answers. It does this by bridging the gap between artificial intelligence and human decision-making processes. Fuzzy logic allows for the modeling of complex systems through a higher level of abstraction, which is derived from our experiential and cognitive knowledge. This is in contrast to classical logic, which requires a profound comprehension of the system at hand. Fuzzy logic does not require a deep dive into the complexities of the system itself.

Quantification that is based on fuzzy logic is understood to be a suitable mathematical approach for evaluating and signifying indeterminacy or ambiguity for a software system that uses fuzzy logic [36]. In contrast to fuzzy logic, which allows for depictions with varied degrees of accuracy, binary logic, sometimes known as "crisp" logic, is only effective with true or false values that are correct. Whenever there is a lack of precision or uncertainty in the information, fuzzy logic almost always plays a constructive role.

This chapter is divided in to six sections:4.1 to 4.6. Section 4.1 introduces the importance of fuzzy logic and its applicability within uncertainties. Section 4.2 explains the method to quantify with the help of fuzzy logic. Section 4.3 discuss about fuzzy operations applying within process of quantification, while section 4.4 discuss about the process of fuzzification and defuzzification. Section 4.5 explain aboutevaluation process with the help of case study; usability expectation of mobile game application for expert user. Section 4.6 concludes with summary of this chapter.

4.2 METHOD FOR QUANTIFICATION USING FUZZY LOGIC

The research presented in this thesis considers quantification via fuzzy logic to be the process of converting qualitative aspects into quantitative assessment. This process contains three fundamental processes that constitute the higher level of abstraction as shown in Fig. 4.1:

1. The procedure of fuzzification.

2. Computationaccording to fuzzy inference system (FIS).

3. The procedure of defuzzification.

The current effort focuses on developing a thorough approach for quantifying data via the use of fuzzy logic.

Step 1: Understand the real-world problem in such a way thatapproaches all the issues and its formulation as per the domain specific mobile application. The decision makers understand the attributes of mobile application along with quality factors as per the current scenario.

Step 2: Carefully design the metrics associated with each quality factors.

Step 3: Define linguistic variables as per fuzzy input and output values, which obtained with the help of fuzzy inference system (FIS).

Step 4: Define fuzzy sets along with membership functions that are appropriate for the mobile application. Apply membership functions to each input and output linguistic variable.Functions such as triangular, Gaussian, and trapezoidal etc. are representative examples of membership functions. This step is achieved with the help of triangular membership function.

Step 5: Assign fuzzy values according to the linguistic variable. For example, there are four scale linguistic variables, H1, H2, H3 and H4 associated with their triangular fuzzy values as shown in Table 4.1.

Table 4.1: Linguistic Variable and Their Fuzzy Values

Lig. Var.	Fuzzy Values
H1	(0, 0, 0.25)
H2	(0.25, 0.375, 0.50)
H3	(0.50,0.625 , 0.75)
H4	(0.75, 1, 1)

Step 6: Aggregate the rules, which involves combining all of the outcomes to generate aggregate fuzzy output value. Methods such as maximum, minimum, and weighted average are used often. This step considers maximum method for aggregation..

Step 7.Perform defuzzification, the process of converting the aggregated fuzzy output values into numeric value (fractional or percentage form). This step considers centroid method as defuzzification technique.

Step 8: Validate the model based on the output in comparison to other methods, rectify the model if divergence exists. Simulate the input and output values, and visualize the whole function.

Step 9: Deploy the numeric outcome as directed by decision makers of mobile application Development Company. Reconsider the model if it does not meet the benchmark.

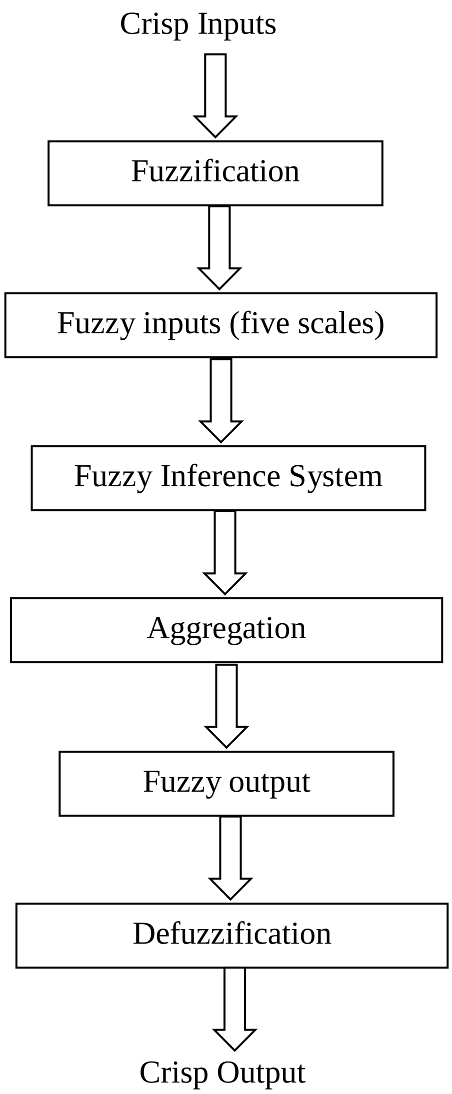


Fig 4.1: Process of Quantification via Fuzzy Logic

4.3 FUZZY OPERATIONS

The present study employs the technique of weighted average for fuzzy sets in order to enhance software quality. In this thesis, the Extension Principle is adopted to carry out the fuzzy operations. The subsequent section elaborates on fuzzy operations [32], including fuzzy addition and fuzzy multiplication.

Fuzzy Multiplication - Assuming (a,b,c) and (x,y,z) are two triangular fuzzy sets, the multiplication of triangular fuzzy sets is mathematically represented as $(a,b,c) \times (x,y,z) = (a\times x, b\times y, c\times z)$.

Fuzzy Addition - Considering (a,b,c) and (x,y,z) as two triangular fuzzy sets, the addition of these fuzzy sets is defined as (a,b,c) + (x,y,z) = [max(a,x), max(b,y), max(c,z)].

4.4 FUZZIFICATION & DEFUZZIFICATION

Fuzzification is the procedure which transforms a crisp input into a fuzzy input. This procedure requires the mapping of a precise input value to a fuzzy set. Fuzzification is a reference to the process. Fuzzy input may takeany value bounded between two crisp inputs. The purpose of fuzzification is to represent the data with such flexibility , not provided by crisp inputs. This is accomplished via the use of fuzzy logic. In order to provide a concrete illustration of this idea, let us take a look at a particular case that involves the fuzzification of recoverable errors as shown in Table 4.2.

Table 4.2: Process of Fuzzification as Per Criteria

Metric	Description	Probability	criteria	Fuzzy rating
Probability (P) to evaluate recoverable errors	$P = \frac{recoverable\ errors}{total\ number\ of\ errors}$	$0 < P < 0.2$	VL (very low)	(0.0 , 0.1 , 0.3)
		$0.2 < P < 0.4$	L (low)	(0.1 , 0.3 , 0.5)
		$0.4 < P < 0.6$	M (medium)	(0.3 , 0.5 , 0.7)
		$0.6 < P < 0.8$	H (high)	(0.5 , 0.7 , 0.9)
		$0.8 < P < 1$	VH (very high)	(0.7 , 0.9 , 1.0)

Defuzzification is the procedure which converts fuzzy output in to crisp output.

There are a few different approaches to defuzzification.

1. Centroid Method (CM)

When it comes to defuzzification, the centroid method is one of the most easy approach presented till now. It determines the center of gravity of the fuzzy set by locating the centroid of the region that is covered by the membership function of the fuzzy set before doing the calculation.

2.Weighted Average Method (WAM)

When this approach is used, each point in the fuzzy set is given a weight is assigned to each point in the fuzzy set that is determined by the degree to which it is a member of the set. After that, the crisp output is computed by taking the weighted average of all of the points that are included inside the fuzzy set.

3. Mean of Maxima Method (MOM)

To calculate MOM, first the maximum values of the membership functions must be determined, and then the average of these maximum values must be calculated.

4. Bisector Method (BM)

To locate the point at which the region that is under the membership function of the fuzzy set is split into two equal halves, The bisector approach is used. The value of bisector point is numeric.

5.Smallest of Maxima Method (SOM)

Using this method, the highest membership value is determined, and the numeric output that corresponds to the rule that smallest of these maxima is chosen.

6. Height Method (HM)

It looks for the point on the y-axis that corresponds to the highest possible membership value and chooses that location.

7. Lambda-Cuts method (LC)

In order to trim the fuzzy set at a certain level, a threshold is considered known as lambda.

The particular properties of the fuzzy logic system and the needs of the application both play a role in determining thedefuzzification. Each approach has its own set of benefits and drawbacks, and the way that is best appropriate for a given situation should be chosen depending on the specific circumstances in which it is used.

The use of the centroid method is utilized in thepresent research study.

$$\text{Centroid formula } z^* = \frac{\int \mu(z).z dz}{\int \mu(z) dz}$$

Where z^* is defuzzified crisp value and $\mu(z)$ is membership function.

4.5 EVALUATION PROCESS

Fuzzy logic is considered to be an effective assessment method for predicting software quality [38] or automatically visualizing it with the help of total metric score [35]. The purpose of quantification is to quantify qualitative aspects in to quantitative assessment. The quantification helps in many ways; developer can compare their assigned quality factors during mobile application development and visualize the weaker area. Tester can compare the same quality factor as per developer but with the different set of metrics and find the gap in between tester's view and developer's view. This section outlines the evaluation process that should be followed in order to accurately assess the overall usability attribute of a domain specific mobile application.

4.5.1 Fuzzy LogicFramework: Design and Evaluation

This section presents following steps to evaluate software quality factors for mobile application with the help of fuzzy logic as:

Step1. Design qualitative aspectsfor the mobile application. The framework depends upon quality factors. Quality factors play a major role and hence optimal numbers of quality factors have to be considered.

Step2. Design a framework for the metrics according to the quality factors undertaken.

Step3. Fuzzy weight is determined by the stakeholders, investors, quality manager or collectively called as decision makers, and fuzzy rates are determined by the developer, tester and β-users in line with prototype of mobile application under development.

Step4. Calculate the overall fuzzy ratings for metrics, quality factors and finally usability expectation for any mobile applicationwith the help of weighted sum method [74]:

$$\textbf{Overall fuzzy rating} = r_1 \times w_1 + r_2 \times w_2 + \ldots\ldots\ldots + r_n \times w_n = \sum r_i \times w_i$$

Step5. The final step is to convert overall fuzzy rating in to crisp valueswith centroid method. Centroid method is one of the well-known methods for defuzzification. Thus, the overall crisp value in terms of percentage for overall quality obtained.

4.5.2 Conversion of qualitative aspects to quantitative assessmentmodel: Case Study

Usability of mobile application refers to the effectiveness and ease of use of a mobile application on a device [27]. Assessment of usability may be extended by existing quality standards [63].

This sectiondiscussed the usability expectation of mobile game for expert users, which is based on the fuzzy logic based evaluation model. The proposed model defines six quality factors, namely Understandability, Memorability, Efficiency, Challengeability, Effectiveness, and User Satisfaction as shown in Table 4.3, to assess the overall quality of a mobile game.Two fuzzy inputs are used in it, namely fuzzy rate and fuzzy weight. The weight is determined according to the priorities of quality factors identified by decision makers, while the fuzzy rate is based on a questionnaire and the experience of expert users for beta version. Both inputs are evaluated using the triangular fuzzy function.

Table 4.3: Usability Attribute and its Quality Factors

Attribute	Quality factors
Usability	Understandability
	Memorability
	Efficiency
	Challengeability
	Effectiveness
	User satisfaction

Proposed model is described in the following steps and illustrated with the help of the diagram as shown in Fig. 4.2.

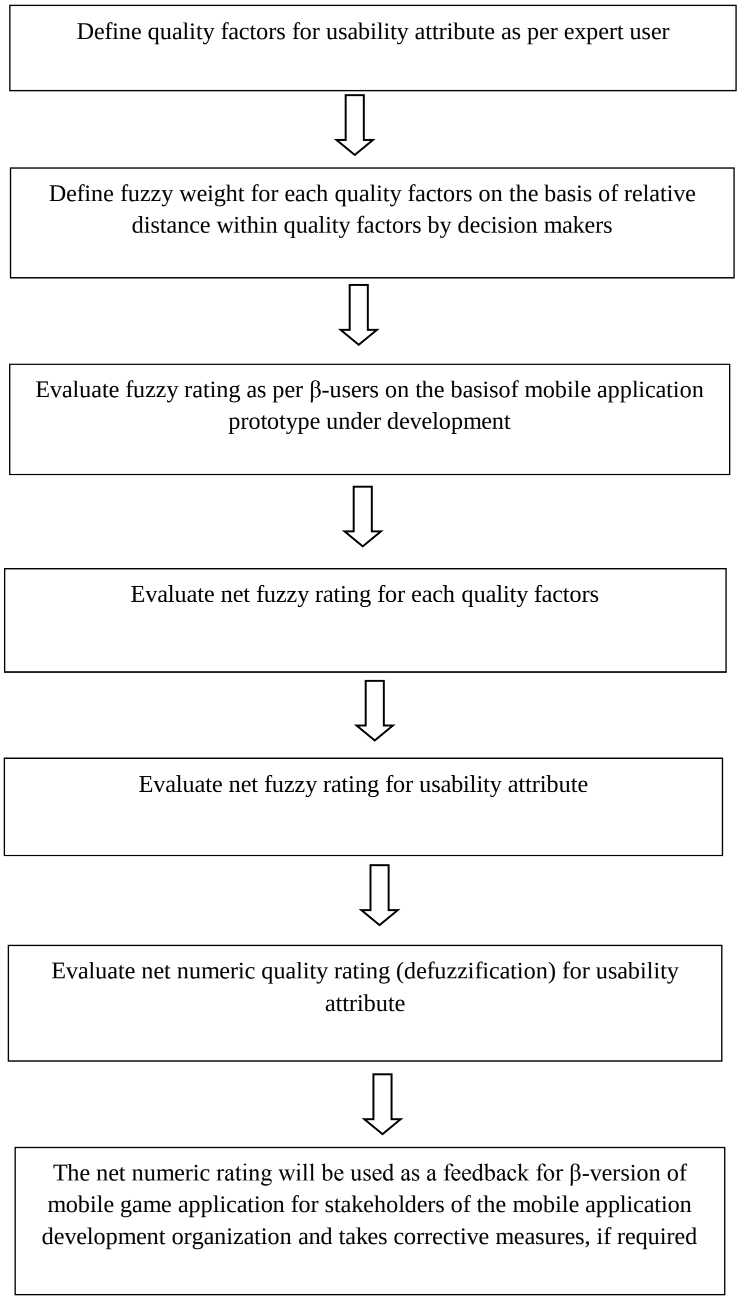


Fig 4.2: Process Flow Diagram for UEM-β

For example a mobile app development company 'ABC' conducts the survey for beta version to evaluate the usabilityas per expert users. This evaluation model has six quality factors, where each quality factor has two inputs; fuzzy rate and fuzzy weight.

All quality factors are evaluated based upon fuzzy values and evaluates overall fuzzy rating for six quality factors which further combined to produce overall quality in fuzzy values. Defuzzification will be applied to convert fuzzy output to crisp output.

The 'ABC' mobile game company adopts five scales, VH(very high), H(high), M(medium), L (low), VL (very low) ratings [45] whose fuzzy values illustrated as Table 4.4.

Table 4.4: Five Scale Fuzzy Weight and Fuzzy Rate

Scale	Fuzzy Weight	Fuzzy Rate
VH	(0.75,1,1)	(0.7,0.9,1)
H	(0.5,0.75,1)	(0.5,0.7,0.9)
M	(0.25,0.50,0.75)	(0.3,0.5,0.7)
L	(0.0,0.25,0.50)	(0.1,0.3,0.5)
VL	(0.0,0.0,0.25)	(0.0,0.1,0.3)

Now evaluating each and every quality factors according to UEM-β:

1. Evaluating the overall fuzzy rating of understandability:

Understandability defined as understanding logical concepts for each task. It is an important quality factor for the experienced user. Metrics designed for understandability and calculate average rating as per average fuzzy rate and average fuzzy weight is calculated as:

Average Fuzzy rate (decided by expert users as per responses of questionnaires): High (0.50, 0.70, 0.90)

Fuzzy Weight (decided by decision makers): Very High (0.75, 1.0, 1.0)

Thus overall rating calculated as:

$$R_{AVG}(\text{Understandability}) = (0.50,0.70,0.90)\text{x}(0.75,1.0,1.0)$$
$$= (0.50\text{x}0.75,\ 0.70\text{x}1.0,\ 0.90\text{x}1.0)$$
$$= (0.38,\ 0.70,\ 0.9)$$

2. **Evaluatingthe overall fuzzy rating of memorability:**

Memorability is defined as the degree of easy with which a user efficiently can remember how to use an application.It is basically trade off in between time and recall pattern, this quality factor specially design for experienced user. Metrics designed for memorability and calculate average rating as per average fuzzy rate and average fuzzy weight is calculated as:

Average Fuzzy rate (decided by expert users as per responses of questionnaires): Medium (0.30,0.50,0.70)

Fuzzy Weight (decided by decision makers): High (0.5,0.75, 1.0)

Thus average rating calculated as:

$$R_{AVG}(\text{Memorability}) = (0.30,0.50,0.70)\text{x}(0.5,0.75,1.0)$$
$$= (0.30\text{x}0.5,\ 0.50\text{x}0.75,\ 0.70\text{x}1.0)$$
$$= (0.15,\ 0.38,\ 0.7)$$

3. **Evaluating the overall fuzzy rating of efficiency:**

Efficiency is defined as ability of user to complete a task with speed and accuracy. This quality factor specially designs for expert user. Metrics designed for efficiency and calculate average rating as per average fuzzy rate and average fuzzy weight is calculated as:

Average Fuzzy rate (decided by expert users as per responses of questionnaires): Low (0.10, 0.30, 0.50)

Fuzzy Weight (decided by decision makers): High (0.5, 0.75, 1.0)

Thus average rating calculated as:

$$R_{AVG}(\text{Efficiency}) = (0.10,0.30,0.50)\text{x}(0.5,0.75,1.0)$$
$$= (0.10\text{x}0.5,\ 0.30\text{x}0.75,\ 0.50\text{x}1.0)$$
$$= (0.05,\ 0.23,\ 0.5)$$

4. **Evaluating the overall fuzzy rating of challengeability:**

Challangeability is defined as to maintain the challenge for every one level increment i.e. form nth level to n+1 th level. This quality factor specially design for experienced user. Matrices designed for challengeability and calculate average rating as per average fuzzy rate and average fuzzy weight is calculated as:

Average Fuzzy rate (decided by expert users as per responses of questionnaires): High (0.50,0.70,0.90)

Fuzzy Weight (decided by decision makers): Very High (0.75,1.0,1.0)

Thus average rating calculated as:

$$R_{AVG}(\text{Challengeability}) = (0.50,0.70,0.90)\text{x}(0.75,1.0,1.0)$$
$$= (0.50\text{x}0.75,\ 0.70\text{x}1.0,\ 0.90\text{x}1.0)$$
$$= (0.38,\ 0.70,\ 0.9)$$

5. Evaluating the overall fuzzy rating of effectiveness:

Effectiveness is defined as ability of user to complete a task in given context. This quality factor specially designs for experienced user. Matrices designed for effectiveness and calculate average rating as per average fuzzy rate and average fuzzy weight is calculated as:

Average Fuzzy rate (decided by expert users as per response of questionnaire): High (0.50,0.70,0.90)

Fuzzy Weight (decided by decision makers): High (0.5,0.75,1.0)

Thus average rating calculated as:

$$R_{AVG}(\text{Effectiveness}) = (0.50,0.70,0.90)\text{x}(0.5,0.75,1.0)$$
$$= (0.50\text{x}0.5,\ 0.70\text{x}0.75,\ 0.90\text{x}1.0)$$
$$= (0.25,\ 0.53,\ 0.9)$$

6. Evaluating the overall fuzzy rating of satisfaction:

The user’s overall level of satisfaction with the system.User’s satisfaction evaluated in terms of feedback. Satisfaction defined as user’s perceived level of expectation and need. Matrices designed for satisfaction and calculate average rating as per average fuzy rate and average fuzzy weight is calculated as:

Average Fuzzy rate (decided by expert users as per response of questionnaire): High (0.50,0.70,0.90)

Fuzzy Weight (decided by decision makers): Very High (0.75,1.0,1.0)

Thus average rating calculated as:

$R_{AVG(}$Satisfaction$_{)}$ = (0.50,0.70,0.90)x(0.75,1.0,1.0)

= (0.50x0.75, 0.70x1.0, 0.90x1.0)

= (0.38, 0.70, 0.9)

Thus Overall rating calculated ($R_{USABILITY}$) as: $R_{AVG(}$Understandability$_{)}$+$R_{AVG(}$Memorability$_{)}$+$R_{AVG(}$Efficiency$_{)}$+$R_{AVG(}$Challengeability$_{)}$+$R_{AVG(}$Effectiveness$_{)}$+$R_{AVG(SATISFACTION)}$

=(0.38,0.70,0.9)+(0.15,0.38,0.7)+(0.05,0.23,0.5)+(0.38,0.70,0.9)+(0.25,0.53,0.9)+(0.38,0.70,0.9)

= (0.38, 0.70, 0.9)

Fuzzy rating obtained as :(0.38, 0.7, 0.9)

Computation of crisp rating (on applying centroid formula):

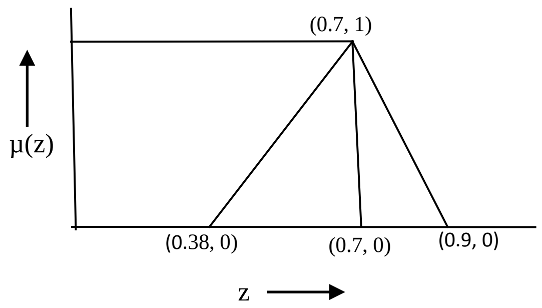


Fig 4.3: Defuzzification using Centroid Formula

Equation of line passing through (0.38, 0) and (0.7, 1)

$\mu(z) = 3.125\,z - 1.1875$

Equation of line passing through (0.9, 0) and (0.7, 1)

$\mu(z) = 4.5 - 5z$

$$Z^* = \frac{\int_{0.38}^{0.7}(3.125\,z - 1.1875)\,z\,dz + \int_{0.7}^{0.9}(4.5 - 5z)\,z\,dz}{\int_{0.38}^{0.7}(3.125\,z - 1.1875)\,dz + \int_{0.7}^{0.9}(4.5 - 5z)\,dz}$$

$$= \frac{0.09493 + 0.07667}{0.16 + 0.1}$$

$$= \frac{0.1716}{0.26}$$

$$= 0.66$$

The crisp value evaluated as 0.66, or66%. Thus the overall quality of mobile application will be calculated as 66%

4.6 SUMMARY

The purpose of this chapter was to offer a fuzzy-based assessment model that transforms qualitative aspects to quantitative assessment. The model was demonstrated by a straightforward case study that assesses the usability expectations of an expert user for a mobile game application by UEM-β. The quantification may be applied in a variety of different ways, such as the technique that is adopted by mobile business companies as an agile software development life cycle (SDLC). By doing so, decision makers of mobile companies will ensure that quality will be approached in the appropriate manner. In the event that corrective steps are required, the quality manager is able to readily examine and compare the various software quality aspects and methodologies being considered. As a result, fuzzy logic is used as a useful tool, which ultimately becomes the fundamental notion for qualitative to quantitative assessment.

CHAPTER 5

PROPOSED MATHEMATICAL FRAMEWORK
MAQM-MA: MULTI ATTRIBUTE QUALITY MODEL FOR MOBILE APPLICATION

5.1 INTRODUCTION

This chapter presents the fuzzy based mathematical evaluation frameworknamed as MAQM-MA (Multi Attribute Quality Model for Mobile Application). This proposed framework MAQM-MA accepts two fuzzy based inputs, fuzzy weight and fuzzy rate. These two inputs have their own impact and they differ with their aim. Multi attribute quality model is a basic framework having qualitative aspects along with multiple attributes. Fuzzy weight will be helpful to determine the relative importance among all software quality factors and thus helps decision makers of any mobile application development company. Fuzzy weight is determined by the decision makers, whereby the pair-wise comparison of software quality factors (SQF) provides a suitable mathematical algorithm on a fuzzy basis, FW-MA (Fuzzy Weight for Mobile Application), which evaluates the relative ranking of the quality factors according to their importance for the quality consideration. Fuzzy rate is an input provided by developer, tester and β-users based on their own questionnaire to calculate the fuzzy rating for the metricsprovided by the decision makers. It becomes a live input represents the prototype of mobile application under development. The combined impact of fuzzy weight and fuzzy rate has to be considered the overall impact of quality and may take the corrective measures during the development of mobile application. The framework MAQM-MA analyzes the qualitative aspects and evaluates it in to quantitative assessmentby considering the perspectives of three different views: the developer, the tester, and the β-user in the form of fuzzy rate along with view of decision maker in the form of fuzzy weight.

The Fuzzy Analytic Hierarchy Process (FAHP) is a decision-making process that uses fuzzy logic and combines qualitative and quantitative methodologies to assess various alternatives and prioritize them, which will be used to determine fuzzy weight

[69].Fuzzy weight also be evaluated with the help of fuzzy least square error approach and fuzzy BWM [48]. The FAHP, introduced by JJ Buckley [40], is an extension of the AHP originally created by TL Saaty [39].

Mobile apps have become an essential component of modern society and have a significant impact on several aspects of our daily routines, including mobile commerce, mobile gaming, and mobile learning. Several researchers have focused on the field of mobile learning. They began by examining the concept and challenges through a survey [56]. They then developed a framework for assessing the quality of mobile learning applications [62, 64]. Additionally, they used Fuzzy TOPSIS to select the best mathematics mobile learning application [65]. Further, they evaluated an e-Reader mobile app [60] and studied the role of usability in game-based learning mobile apps [61].Researchers also provide their perspective on the factors contributing to the success of mobile commerce applications [70], as well as the design of interfaces to enhance usability [58]. They also examine usability expectations using the GQM paradigm and ISO/IEC-9241-11 quality standard [68].

This chapter divided in to eight sections 5.1 to 5.8. Section 5.1 introduces about FW-MA algorithm and fuzzy based mathematical framework MAQM-MA. Section 5.2 explains the concept about proposed framework MAQM-MA. Section 5.3 discuss about designing of MAQM-MA framework.Section 5.4 discussedthe quality factors and metrics along with evaluation criteriafor fuzzy rating as per developer's view. Section 5.5 discussed the quality factors and metrics along with evaluation criteriafor fuzzy rating as per tester's view. Section 5.6 discussed the quality factors and metrics with the help of GQM paradigmas per β-user's view. Section 5.7 presents flow graph for over-all view of MAQM-MA. Section 5.8 concludes with summary of this chapter.

5.2 CONCEPT OF PROPOSED FRAMEWORK: MAQM-MA

This section proposed a fuzzy logic based mathematical framework MAQM-MA, which is used to evaluate overall quality of a mobile application. The observation depends upon three views developer, tester and β-user during the development of mobile application. The purpose of fuzzy based mathematical framework to evaluate and fulfill the purpose to convert qualitative aspect to quantitative assessment. The overall quantifiable quality depends upon combination of quantification of three views. Quantification depends upon two inputs, fuzzy rating and fuzzy weight. Fuzzy rating: an input which depends upon feedback of prototype of mobile application during its development. This input has five scale fuzzy rating input as Very Low, Low, Medium, High and Very High. Fuzzy weight depends upon relative importance among quality factors with a constraint such that sum of weights of quality factors will be one. Relative importance of quality factors is decided by the stakeholders, investors, quality manager collectively called as decision makers.

A mobile application development organization mayconsist of stakeholders, investors, quality managers and customer or collectively called as decision makersalong with domain specialist beta-users as well as technical or non-technical persons as β-user tester, all of them are viewedas an asset for any mobile application business.

Purpose of fuzzy weight and fuzzy rate serve as:

Fuzzy weight of quality factor: The fuzzy weight is defined according to the relative significance of the quality factors and this will be assessed by decision makers or a subset of decision makers via pair-wise comparisons.

Fuzzy rate of quality factor: The fuzzy rate will be assessed by the developer, the tester, and the β-users based on their feedback as per questionnaires or checklists that are directed by the decision makers and guided by the software prototype.

5.3 DESIGN OF PROPOSED FRAMEWORK MAQM-MA

The proposed frameworkincorporates some of the features of two well-known quality standardsISO/IEC-9126 and ISO/IEC-9241-11. The baseline standard referred to as ISO/IEC-9126 is used in this study for developer and tester’s view and ISO/IEC-9241-11 for β-user’s view. Accordingly, the quality elements that need to be taken into consideration for the developer, the tester, and the β-user are those that are based on ISO/IEC-9126 quality criteria. During the assessment, the fuzzy rating and fuzzy weight of each and every metric that is related with its quality component as indicated by ISO/IEC-9126 are taken into consideration. Obtaining the overall fuzzy rating of a specific quality factor is accomplished by combining two fuzzy operations, namely fuzzy multiplication of fuzzy weight and fuzzy rating of particular metric. These fuzzy operations are then combined with the assistance of fuzzy addition operation, which ultimately results in the overall fuzzy rating of a particular quality factor. The same procedure is carried out in order to achieve the overall rating from the perspective of the developer, the tester, or the β-users.

This research study takes into consideration the following quality factors from the perspectives of the developer and tester. Customization and data availability are two additional quality factors that have been added to the list of quality factors that have been extracted from quality standardISO/IEC-9126.All those quality factors that are to be considered marked with ✓ (tick) as illustrated in table 5.1. Quality standardISO/IEC-9241-11 considered for usability, recognized world-wide. All of the quality factors are taken to be considered for β-users, marked with ✓ (tick) as illustrated in table 5.2.

Table 5.1: Quality Factors for Developer and Tester according to ISO/IEC-9126

Attributes	**Quality-Factors**	**Developer**	**Tester**
Functionality	Suitability	✓	✓
	Accuracy	✓	✓
	Interoperability	✓	✓
	Functionality Compliance	✓	✓
	Security	✓	✓
	Customization-new	✓	✓
Efficiency	Time Behavior	✓	✓
	Resource Behavior	✓	✓
	Efficiency Compliance	✓	✓
	Scalability	✓	✓
	Performance	✓	✓

	Data-availability-new	✓	✓
Portability	Replace-ability	✓	✓
	Adaptability	✓	✓
	Install-ability	✓	✓
	Co – existence	✓	✓
	Portability Compliance	✓	✓
Maintainability	Analyzability	✓	✓
	Changeability		✓
	Testability		✓
	Stability	✓	✓
	Maintainability Compliance	✓	✓
Reliability	Maturity		✓
	Fault Tolerance	✓	✓
	Recoverability	✓	✓
	Reliability Compliance	✓	✓
		23	26

Table 5.2: Quality Factors for β-User according to ISO/IEC-9241-11

Attribute	Quality factors	β-Users
Usability	Feature	✓
	Time taken	✓
	Learn ability	✓
	Accuracy	✓
	Security	✓
	B-user's feedback	✓

5.3.1 FW-MA algorithm to evaluate prioritization of fuzzy weight for developer, tester and β-userperspectives. The proposedalgorithm is based on the procedure given byNădăban&Dzitac[73]:

Step1. Design the qualitative framework for any mobile application.

Step2.Decide pair-wise priorities (crisp value) among software quality factors as per decision maker.

Step3.Replace crisp value with corresponding fuzzy values accordance as linguistic variable such as equally imp, strongly imp etc.

Step4.Calculate fuzzy geometric mean (FGM) for all quality factors (QF).

Step5.Calculate normalized fuzzy weight (NFW) from fuzzy geometric mean (FGM) for all software quality factors (SQFs).

Step6.Convert normalized fuzzy weight (NFW) in to corresponding crisp value and find the relative distance in between quality factors.

Step7. Assign fuzzy weight by decision maker as per relative distance in between the quality factors.

5.3.2 Flow graph of FW-MA algorithm

Flow graph of the FW-MA algorithm, which evaluates the relative ranking of the software quality factors (SQFs) according to their importance for the quality assessment as shown in Fig 5.1.

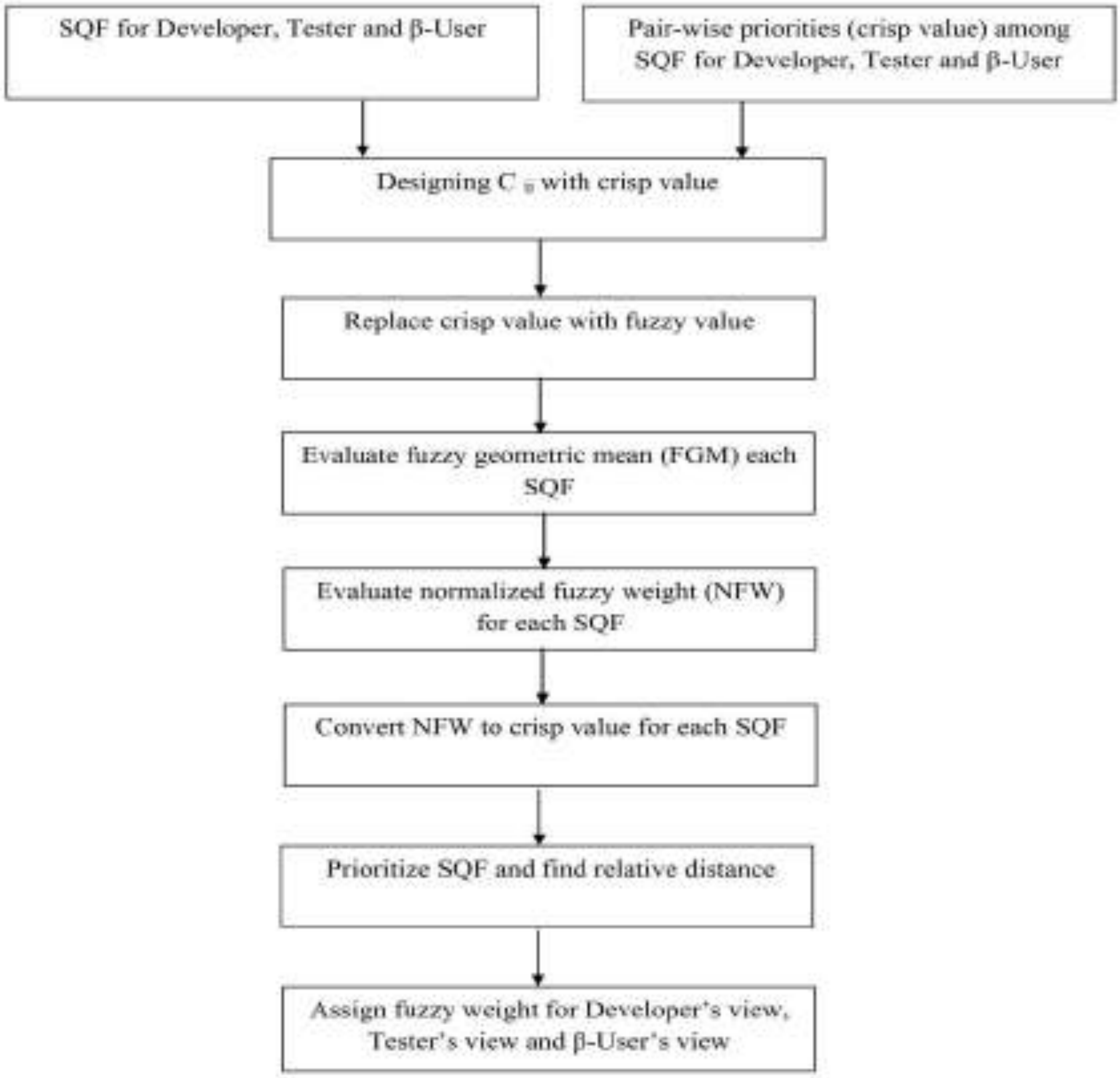


Fig. 5.1: Flow graph of FW-MAAlgorithm

5.3.3MAQM-MAframework to evaluate overall quality (crisp):

Step1. Allocation of appropriate fuzzy weight for each metrics as per decision makers.

Step2. Allocation of appropriate fuzzy rate accordance as evaluation criteria for each metrics by developer, tester and β-users provided by decision makers.

Step3. Evaluate appropriate fuzzy rate for each quality factors with the help of fuzzy weight and fuzzy rate of metrics.

Step4. Establish the priority with in quality factors with pair-wise comparison.

Step5. Allocation of appropriate fuzzy weightfor each quality factor accordanceasthe priority among quality factors.

Step6. Evaluate overall fuzzy ratefor developer, tester and β-users with the help of fuzzy weight and fuzzy rate of quality factors .

Step7. Evaluate overall quality (crisp) for developer, tester and β-users with the help of defuzzification.

5.4 DEVELOPER'S VIEW

According to the developer, the qualitative aspect of a mobile application depends on the software quality factors that are associated with the developeralong with the metrics. The following section5.4.2illustrates the relevant quality factors, as well as the metrics and the evaluation technique for fuzzy input rating as shown in Table 5.3. One of the issues that concerns stakeholders, investors, quality managers, and all thosereferred to as decision makers, is the appropriateallocation of fuzzy weight. An additional fuzzy-based mathematical algorithm known as FW-MA, thatusespair-wise comparison to prioritize quality factors.

5.4.1Flow graph of MAQM-MA (Developer's View)

Flow graph of the MAQM-MA framework, which evaluates the overall crisp quality according to the developer as shown in Fig 5.2.

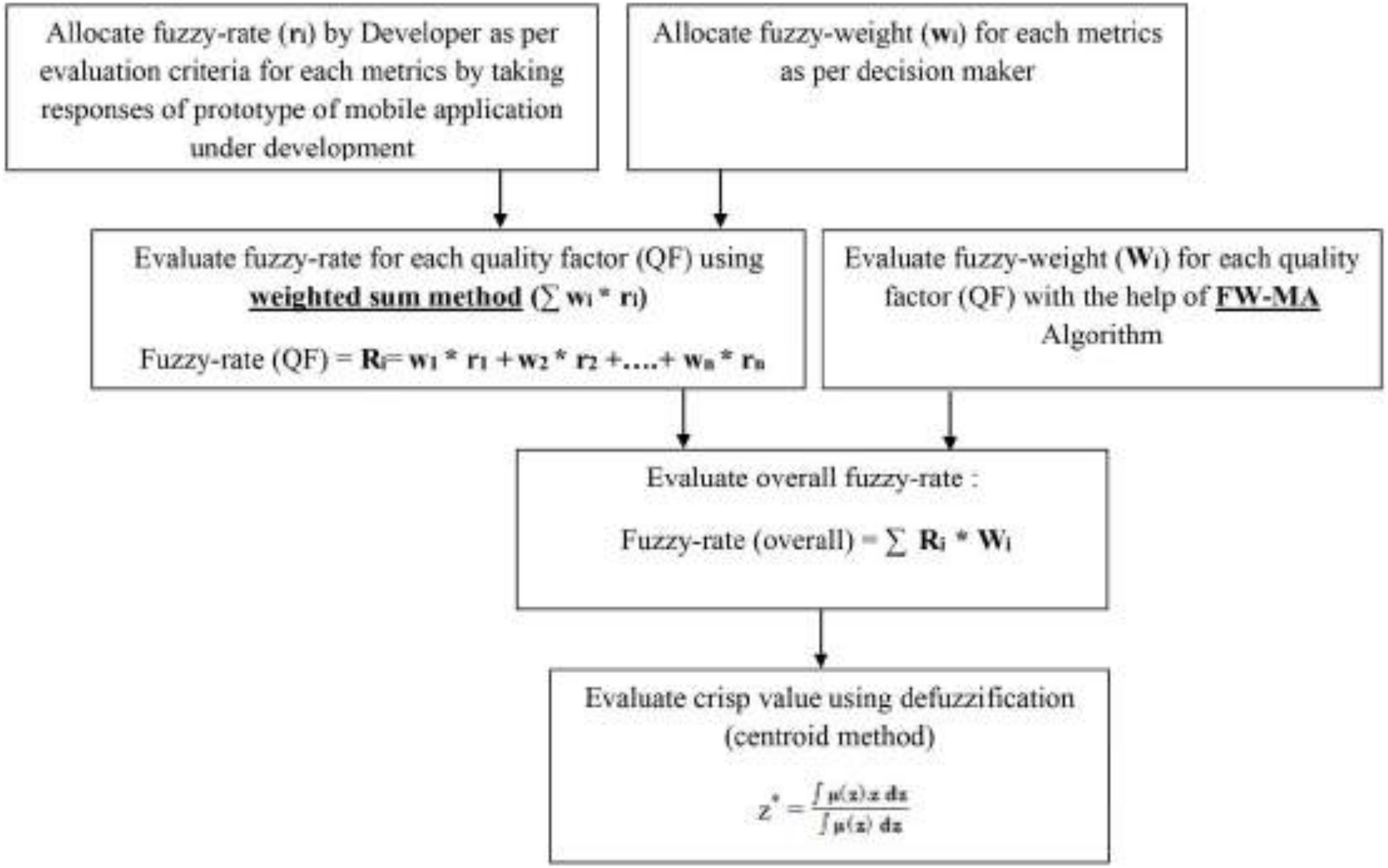


Fig. 5.2: Flow graph of MAQM-MA: Developer's View

5.4.2Assessing fuzzy rating for each metric: developers view

Table 5.3: Evaluation criteria for assigning fuzzy rating according to developer

Quality factor	Metrics	Evaluation Criteria	Fuzzy Parameter
Suitability	M1: Find P for operations implemented.	(P) Probability = $\frac{\text{total no of implemented operations}}{\text{no of operations as per document provided to developer}}$	If p=1 then VH otherwise L
Accuracy	M1: Find P for accurate results for each sensor.	(P) Probability = $\frac{\text{total no of operations producing accurate result for each sensor}}{\text{total no of operations for each sensor}}$	If p=1 then VH otherwise L
	M2: Find P for accurate results for multiple sensors.	(P)Probability= $\frac{\text{total no of operations producing accurate result for each multiple sensor}}{\text{total no of operations for each multiple sensor}}$	If p=1 then VH otherwise H
Interoperability	M1: App behaves well for all H/W& OS.	Is app behaves well in both the platforms as per combination of H/W and OS (android or IOS)?	If true then VH otherwise L
	M2: Behavior of each API functions.	Is app behaves well as per API functions in both the platforms as per combination of h/w and OS (android or IOS)?	If true then VH otherwise L
	M3: Behavior of each	Is app behaves well as per sensors in both the platforms as per combination	If true then VH

	sensor.	of h/w and OS (android or IOS)	otherwise H
Functionality Compliance	M1: Adherence to functionality standard	Feedback as per developer	VH to VL
Security	M1: Degree of access controllability.	appropriate level of abstraction with: 1. Biometric 2. OTP based 3. Login / Password	Biometric (VH) OTP based (M) login/password (L)
	M2: Mode of payment gateway.	Most trusted and popular payment gateway.	VH, H, L
Customization-new	M1: Display as per device screen.	Overall content of application including buttons displayed as per device screen.	VH,M,L
	M2: Font size and style as per UI provided by client.	Font size and style as per UI provided by client and feedback during agile development.	VH,H, M
	M3: About landscape and portrait.	Support landscape and portrait mode as per size of screen.	VH, L
	M4: Fill a complete form	Optimized to fill a complete form as per size of screen.	VH, M, L

	as per size of screen.		
Time Behavior	M1: About response time	Compare response time with server emulator and different devices. Response time = Application Delay + H/W Delay + N/W Delay	If time is same VH otherwise H, L
	M2: Throughput Rate	How many requests executed in unit time?	VH to VL
Resource Behavior	M1. Memory and processor as per app M2.H\W selection M3.Tradeoff in between H/W selection and power dissipation. M4.Duration of execution of app and power dissipation. M5. N\W selection and execution of app	1.memory and processor as per app 2.H/W selection as per domain of mobile app 3. Tradeoff in between combination of H/W and power dissipation. 4. Tradeoff in between duration of execution of app and power dissipation. 5. tradeoff in between n\w selection and execution of app	VH to VL VH to VL VH to VL VH to VL VH to VL
Efficiency Compliance	M1: Adherence to efficiency standard	Overall feedback as per developer	VHto VL

Scalability	M1. Support for multiple user. M2. What if users randomly increased? M3. Mobility Test	1. Support for multiple user. 2. What if users randomly increased. 3. Test overall performance as per N\W selection in lab and outside the lab	VH to VL VH,H, L VH to V L
Performance	M1.Optimal threading model to avoid suspension or shut down of thread. M2.Algo to implement priority among threads.	1. App does not stop functioning. 2. Check priority wise application.	if yes VH otherwise VL if it is behaves properly then VH otherwise VL
Data-availability-new	M1.App with data to display initially, even if low or no connectivity. M2.Use of Caching scheme and local storage of data. M3.Special care about data dependent application.	1. App installed first time with data even no connectivity. Then VH otherwise H 2. Optimal utilization of caching and local storage of data. 3. Data optimization in case of data dependent application.	VH, H VH, H, M VH, H, M

Replace-ability	Opportunity to adapt app in android as well as IOS.	Response time almost same in both platform if yes then VH , time is average then M, time too long VL	VH, M , VL
Adaptability	1. Optimal combination of H/W for all devices and for all platforms.	Check response time for each platform if yes then VH,M,VL	VH, M , VL
Install-ability	Relative ease of installing software for a given platform	Compare time to install app for both platform,android as well as IOS	VH, M, VL
Co – existence: ability to exist with other independent software	Frequency of deadlocks	If frequent occurs then VL otherwise ,M,VH	VH, M, VL
Portability Compliance	M1: Adherence to Portability standard	Overall feedback as per developer	VH TO VL
Analyzability	M1.KLOC (size oriented metric)	1.KLOC / team member VH to VL	VH to VL

	M2.Skills M3.Experience M4.Documentation M5. CMM applicable.	2.Skills:exp in that software skill: VH to VL 3.Experience:overall exp VH to VL 4.Documentation Easy documentation, understood by other developer. 5. CMM applicable (level 1 to 5). VH to VL	VH to VL VH to VL VH to VL VH to VL
Stability	M1. No of global variables. M2.Support error handling or exception handling.	1.If lesser no of global variable then VH else M , VL 2.Exceptional handling for all kind of errors then VH	VH , M, VL VH , H, L
Maintainability Compliance	M1: Adherence to Maintainability standard	Overall feedback as per developer	VH TO VL
Fault Tolerance	M1: Specific level of performance M2: Performance after	1.If exception handling then VH else L 2.P= no of successfully updated functions/no of fault prone functions Where P is probability of functions successfully met after updation..	VH, L VH TO VL

	updation.	If P=0 VL P = 0.3 L P = 0.6 M P = 0.8 H P = 1 VH	
Recoverability	M1: Degree of recoverability.	If data backup provided by nearest cloud then VH else L	VH, L
Reliability Compliance	M1: Adherence to Reliability standard	Overall feedback as per developer	VH TO VL

5.5 TESTER'S VIEW

According to the tester, the qualitative aspect of a mobile application depends on the software quality factors that are associated with the tester along with the metrics. The following section 5.5.2 illustrates the relevant quality factors, as well as the metrics and the evaluation technique for fuzzy input rating as shown in Table 5.4. An additional fuzzy-based mathematical algorithm known as FW-MA, that uses pair-wise comparison to prioritize quality factors.

5.5.1 Flow graph of MAQM-MA (Tester's View)

Flow graph of the MAQM-MA framework, which evaluates the overall crisp quality according to the Tester as shown in Fig 5.3.

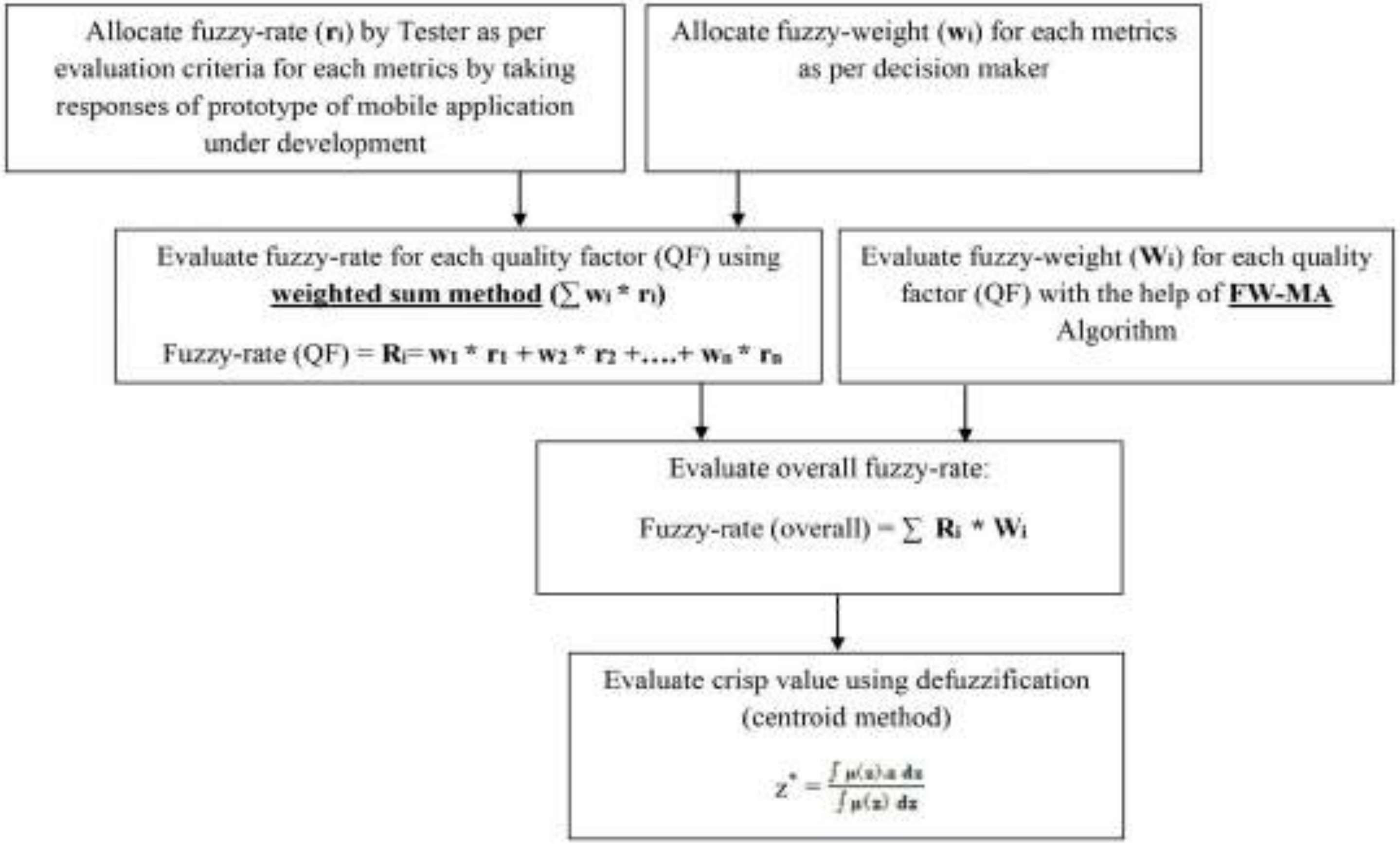


Fig. 5.3: Flow graph of MAQM-MA :Tester's View

5.5.2 Assessing fuzzy rating for each metric: tester view

Table 5.4: Evaluation criteria for assigning fuzzy rating according to tester

Quality factor	Metrics	Evaluation criteria	Fuzzy parameter
Suitability	M1: Implementation of all functions.	Check all the functions are implemented or not. If all implemented then VH	VH, H, M
Accuracy	M1: Behavior of sensors	Does each operation well execute as per input wrt sensors .accurately then VH	VH, H, M
	M2: Precision of computation	All computation performs up to required precision then VH	VH, H, M
Interoperability	M1. Easily installed first time for android and ios.	If easily installed then VH	VH, H, M
	M2. Uninstalled, then again installed easily for android and ios.	If easily installed then VH	VH, H, M
	M3. Behavior of multiple sensors.	If multiple sensors behave well for android and ios.	VH , M , L

		then VH	
Functionality Compliance	M1: Adherence to Functionality standard	Overall feedback as per Tester	VH TO VL
Security	M1:Degree of access controllability,	Test for appropriate level of abstraction with: (Biometric)VH, (OTP based)H (login/password)M	VH, H, M
Customization-new	M1: Overall content of application including buttons displayed as per device screen. M2: font size and style. M3:Landscape and portrait mode M4: Optimized to fill a complete form as per size of screen.	1. Test overall content of application including buttons displayed as per device screen. If it is displayed for all screen then VH 2. Test font size and style as per UI provided by client and feedback during agile development. then VH, H, M 3. is it support landscape and portrait mode then VH 4. Is it Optimized to fill a complete form as per size of screen? Then VH	VH, H, M VH, H, M VH, M VH, H, M
Time Behavior	M1: Response time as per devices and	1 if there is feasible (expected) response time for	VH, H, M

	network.	allconditions then VH	
Resource Behavior	M1:App behavior as per battery. M2: Level of battery as per duration.	1. if app behaves well even if battery is low. VH to VL 2. Is level of battery reduced drastically as per duration of execution of app?	VH to VL VH to VL
Efficiency Compliance	M1: Adherence to efficiency standard	Overall feedback as per Tester	VH TO VL
Scalability	M1: Support multiple users M2: Load of users M3: Mobility test	1. Support for multiple user. 2. What if users randomly increased? 3. Test for N\W selection in lab and outside	VH to VL VH,M,L VH,M,VL
Performance	M1: loading time for app. M2: Functioning when switch to other app. M3: Phone calls do not create obstacle in functioning of app.	1. Loading time for app not to be too long. If yes then VL otherwise H 2. App does not stop functioning when switch to other app. If yes then VL 3. Phone calls do not create obstacle in functioning of app. If yes then VH 4. Check response time under various devices and	VL, H VL, M, H VH, M, VL VH, M, L

	M4:Response time as per devices and n\w.	n\w selection. If time is feasible then VH	
Data-availability-new	M1: Data available in first time.	Is data available in first time even if connectivity is low or no connectivity? If yes then VH otherwise M	VH, M
Replace-ability	M1: Updation in app.	Is easily update to its new version if yes VH	VH, H, M
Adaptability	M1:Response time for each platform	Check response time for each platform if yes then VH	VH, H, VL
Install-ability	M1: Install formsource in both platform.	Compare time to install app with in both platform.	VH,M,VL
Co – existence	M1:App hangs for each platform.	Check app hangs for each platform or not, if frequent then VL	VH to VL
Portability Compliance	M1: Adherence to portability standard	Overall feedback as per Tester	VH TO VL
Analyzability	M1: Easy to understand mobile prototype	Easy to understandall functional and non-functional aspects	VH to VL
Stability	M1: Degree of stability	If anyone attempts for unexpected input? Proper exception handling or not?	VH to VL
Maintainability Compliance	M1: Adherence to maintainability standard	Overall feedback as per Tester	VH TO VL
Fault Tolerance	M1:Breakdown avoidance	X=1-(A/B) A is no of breakdown	VH to VL

		B is no of failures If X=0 VL X=0.3 L X=0.6 M X=0.8 H X=1 VH	
Recoverability	M1: Degree of recoverability	Feasible time for recoverability	VH to VL
Reliability Compliance	M1: Adherence to reliability standard	Overall feedback as per Tester	VH TO VL
Changeability	M1.Updated versions M2. Removal of errors	1.Easy to modify as per updated versions 2.Easy to modify as per removal of errors	VH TO VL VH TO VL
Testability	M1: Optimal no of test cases. M2: Cyclomatic complexity. M3: Kinds of test. M4: Manual and automated testing.	1.Test cases are optimal 2. If CC<10 then VH 3. (functional, performance, API , usability etc.) have to be performed. 4.Perform both testing	VH, M, L VH, H, M VH TO VL VH, M
Maturity	M1:Failure density against test cases	1.X=A1/A2 A1 is no of failures A2 is no of test cases If x=0 VH	VH TO VL

	M2:Failure resolution	X=.3 H X=.6 M X=.8 L X=1 VL 2. Y=A1/A2 A1 is no of failure resolved A2 is total no of failure detected If Y=0 VL Y=0.3 L Y=0.6 M Y=0.8 H Y=1 VH	VH TO VL

5.6 β-USER'S VIEW

The β-user's perspective is focused on the usability factor. The baseline standard for usability is derived from ISO/IEC-9241-11, the most recent quality standard for usability, which incorporates the suggested framework MAQM-MA. Section 5.6.2 illustrates the usability factors and metrics for M-Commerce application according to ISO/IEC-9241-11 along with GQM (Goal Question Metrics) approach [23], to access fuzzy rating as shown in Table 5.5. An additional fuzzy-based mathematical algorithm known as FW-MA, that uses pair-wise comparison to prioritize quality factors.

5.6.1 Flow graph of MAQM-MA (β-User's View)

Flow graph of the MAQM-MA framework, which evaluates the overall crisp quality according to the β-User'sas shown in Fig 5.4.

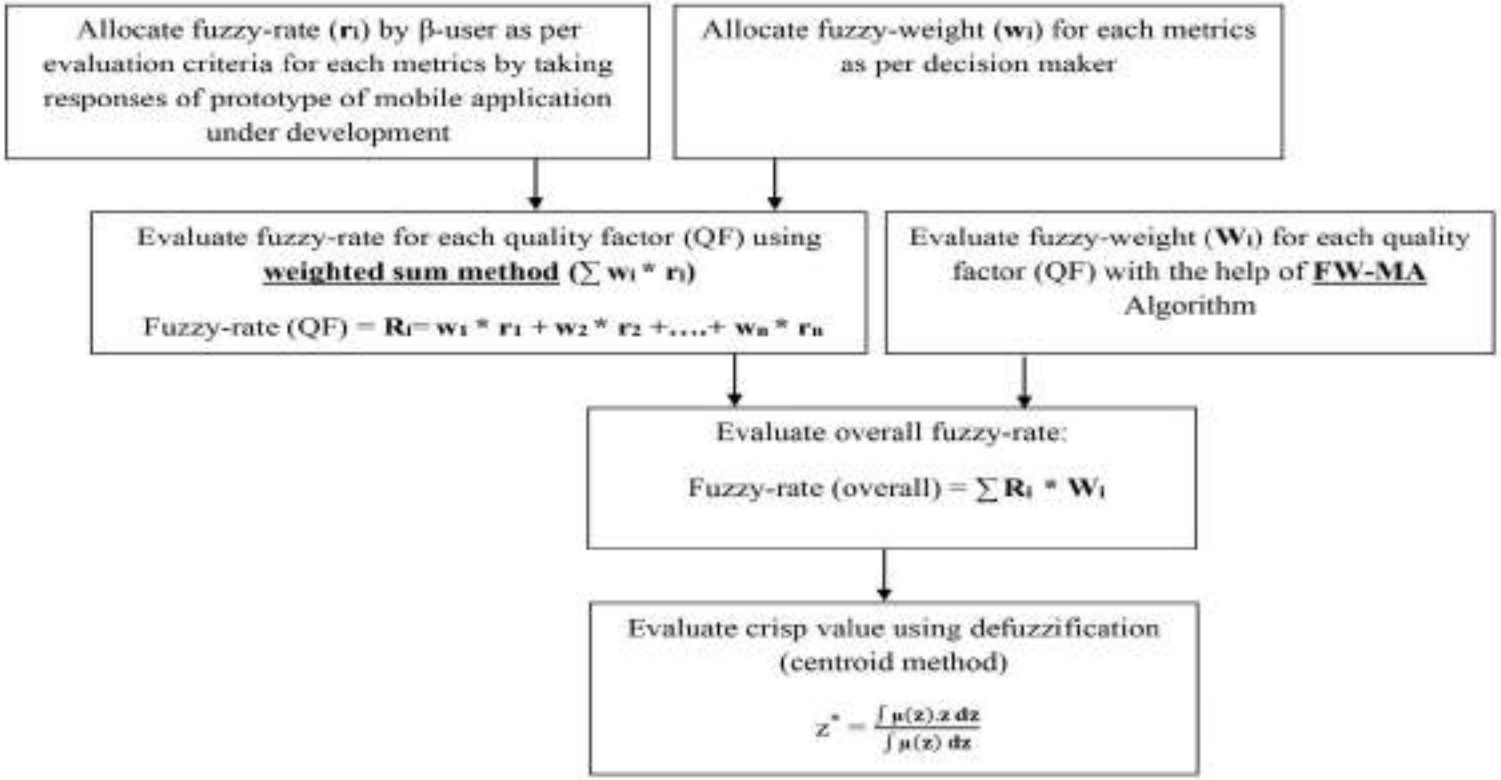


Fig. 5.4: Flow graph of MAQM-MA :β-User's View

5.6.2Assessing fuzzy rating for each metric: β-User's view:

Table 5.5: Evaluation criteria for assigning fuzzy rating according to β-User's

Usability factor	Goal	Question	Metric	Fuzzy Parameter
Efficiency	Features	About display & navigation?	Impact of text, font and colors	VH to VL
			Degree of navigation in App	VH to VL
			Suitable for night vision	VH to VL
		What about search pattern (text pattern and speech pattern)?	Finding information about product are easier	VH to VL
			Speech to text converted accurately	VH to VL
		What about product description?	About description of product	VH to VL
			Display of product photograph	VH to VL
			About display of product price	VH to VL
			Display of product status (available or out of stock)	VH to VL
		What about purchase process?	Easy to register	VH to VL
			Easy to change customer information	VH to VL
			Easy to order product	VH to VL
			Shopping cart's information is accurate	VH to VL
			Adequate information about how to order	VH to VL
			Adequate information about payment options	VH to VL

			Adequate information about how to cancel the product	VH to VL
			Adequate information about return & refund policy	VH to VL
			Adequate information about order detail	VH to VL
			Adequate information about delivery time	VH to VL
			Adequate information about delivery cost	VH to VL
			Adequate information about delivery area	VH to VL
			Delivery to other address	VH to VL
			Online order tracking available	VH to VL
	Time taken	What about time to search a sub-task?	Search particular product	VH to VL
		What about time to complete a whole task?	Time to complete a task	VH to VL
Effectiveness	Learn-ability	What about intuitive learning about mobile app?	Easy to learn interface	VH to VL
			Adequate content management	VH to VL
		What about help or customer service?	Adequate help (demo version)	VH to VL
			Adequate help (text version)	VH to VL
	Accuracy	What about Mobile app response as per action?	Mobile app respond properly as per action	VH to VL
			Every component of interface respond accurately	VH to VL

		What about chance for successful completion of task or sub-task?	Probability to search successfully task completion in 1st attempt	VH to VL
			Probability to completion of task within given time	VH to VL
Satisfaction	Security	What about security of personal data & financial data?	Adequate information about privacy policy	VH to VL
			Secure socket layer used by mobile app	VH to VL
			Well recognized secure payment methods	VH to VL
			Different mode for verification such as OTP based	VH to VL
	Feedback	What about feedback process?	Overall features	VH to VL
			Overall learning process	VH to VL
			Overall accuracy	VH to VL
			Overall security	VH to VL
			Overall experience to use this mobile app	VH to VL

5.7 MAQM-MA (OVER-ALL VIEW): FRAMEWORK

Flow graph of the MAQM-MA framework, which evaluates the overall crisp quality as shown in Fig 5.5.

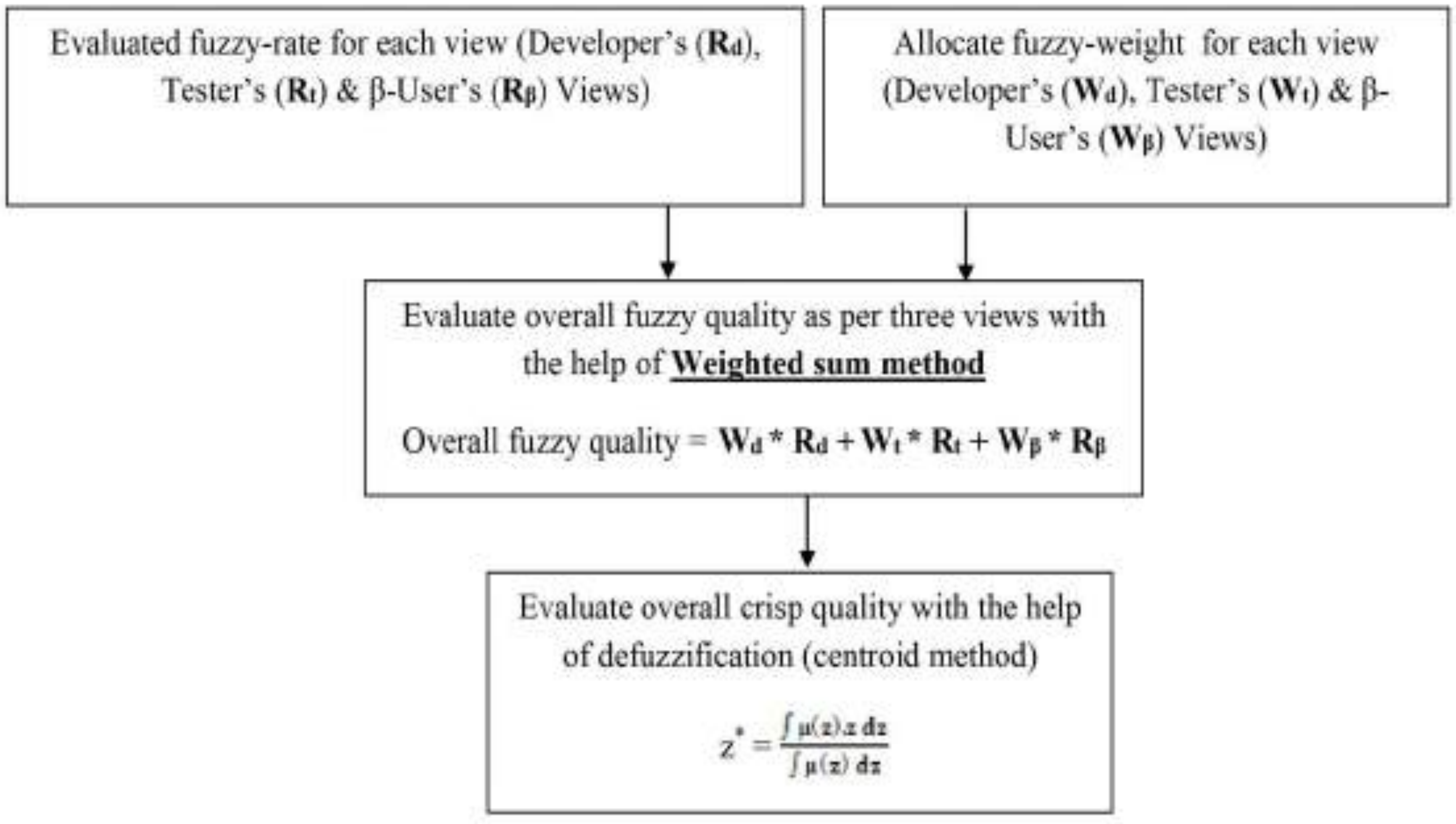


Fig. 5.5: MAQM-MA (Over-all View): Framework

5.8 SUMMARY

A generic fuzzy-based assessment framework known as MAQM-MA is presented in this chapter. Thisframework includes a total of 23 software quality factors and 46 metrics specific to developers, 26 SQFs and 43 metrics specific to testers, six SQFs and 42 metrics specific to β-users. This framework considers three different perspectives developer, tester and β-users. Two of the perspectives, developer and tester, share some common quality factors as per ISO/IEC-9126 quality standard, but they assess the quality by using distinct sets of metrics. The β-user's perspective is focused on the usability factor, derived from ISO/IEC-9241-11, the most recent quality standard for usability.

CHAPTER 6

QUANTIFICATION OF QUALITY FACTORS USING MAQM-MA FRAMEWORK

6.1 INTRODUCTION

Software quality factors represent an specific software quality attribute. Quantifying quality factors involves measuring a specific attribute. Certain software products may possess a specific attribute that holds great significance. In such cases quantification have tried to evaluate software quality attributes reliability [26], reliability of aspect oriented software [33] and maintainability [34]. ISO/IEC-9126 standard can be extended to evaluate source code attribute [18]. Quantification of software quality factor can also be evaluated bytop-down approach [16].

Fuzzy multi-criteria decision making (FMCDM) employs fuzzy logic to solve MCDM issues using vague or unclear data and may use for the selection of software development strategy [44] . FMCDM represents a promising tool for addressing decision making in intricate problem domains [41]. Similar analysis has been done using the ISO/IEC-9126 quality standard [46, 47]. An aggregation method choquet integral, used to solve problem of fuzzy multi criteria decision approach [43], or by using the rough sets and fuzzy sets [42].

This chapter presents the quantification of fuzzy based mathematical framework MAQM-MA, discussed in chapter five, with the assistance of a case study that is based on an M-Commerce application. The scope of this case study involves three viewpoints: the developer view, the tester view, and the β-user view. Each of these perspectives has its quality factors along with own set of metrics and quality considerations. In this case study, two fuzzy inputs, fuzzy weight and fuzzy rate are taken into consideration. The fuzzy weight of the input is considered to be implicit, and it is taken into consideration by the investors, stakeholders, software quality management, or collectively called as decision makers in the mobile application industry. The fuzzy weight of the input was assessed according to relative importance of quality factors to the concerns that were shared and mutually agreed by the decision makers. During the process of developing a mobile application, the input fuzzy rate was analyzed in accordance with the prototype of the mobile application under development. MAQM-MA combines two quality standardsISO/IEC-9126 and ISO/IEC-9241-11. ISO/IEC-9126 quality standard considers for developer and tester viewwhile ISO/IEC-9241-11 considers for β-user, because it is specially meant for usability.

This chapter divided in to five sections 6.1 to 6.5. Section 6.1 defines quantification process for the fuzzy based mathematical framework MAQM-MA. Section 6.2 explains the process of quantification of MAQM-MA framework for three perspectives i.e. Developer, tester and β-user, as a result obtained overall fuzzy rating.Section 6.3 explains about process of quantification as anoverall effect of three perspectives. Section 6.4 discussed the process of defuzzification and obtained overall crisp quality as per three perspectives and their overall effect. Section 6.5 concludes with summary of this chapter.

6.2 QUANTIFICATION OF SOFTWARE QUALITY FACTORS USING MAQM-MA FRAMEWORK

In the case study of the mobile commerce application, there are three perspectives that are taken into consideration, and each perspective requires two fuzzy inputs: fuzzy weight and fuzzy rate. When making decisions about mobile businesses, fuzzy weight should be taken into consideration by the decision makers, and fuzzy ratings should be reviewed according to the prototype for each perspectives. The overall quality of mobile application depends upon quantification as per three perspectives and their combined effect.

6.2.1 Quantification of quality factors using MAQM-MA framework :Developer's view

The developer has a total of 46 metrics, MD1 to MD46 and 23 quality factors, QFD1 to QFD23. Fuzzy weight and fuzzy rate are the two inputs that are used for each measure. This calculation adopts algorithm as discussed in sections5.3.1 and 5.3.3 of chapter 5. Fuzzy rating for each quality factor was examined with the assistance of both fuzzy inputs that were given for metrics, and as a result, an overall fuzzy rating of the developer viewpoints was generated with the help of fuzzy weight and fuzzy rating of quality factors.

The following steps will illustrate the evaluation:

Step1. Evaluate overall quality (fuzzy rating) for quality factors QFD1 to QFD23 with the help of fuzzy weight and fuzzy rate assigned for each metrics as shown in Table 6.1. The questionnaire facilitates the assignment of fuzzy rates for the metrics (APPENDIX-1).

Table 6.1: Evaluation of Fuzzy Rating for each QF: Developer's View

	Quality Factors	Fuzzy Rating(Metrics)					Fuzzy weight(Metrics)				Fuzzy Rating (Quality Factors)		
QFD1	Suitability	MD1	VH	0.7	0.9	1	VH	0.75	1	1	0.525	0.9	1
QFD2	Accuracy	MD2	VH	0.7	0.9	1	VH	0.75	1	1	0.525	0.9	1
		MD3	H	0.5	0.7	0.9	VH	0.75	1	1			
QFD3	Interoperability	MD4	VH	0.7	0.9	1	VH	0.75	1	1	0.525	0.9	1
		MD5	VH	0.7	0.9	1	H	0.5	0.75	1			
		MD6	H	0.5	0.7	0.9	H	0.5	0.75	1			
QFD4	Functionality Compliance	MD7	H	0.5	0.7	0.9	H	0.5	0.75	1	0.25	0.525	0.9

QFD5	Security	MD8	VH	0.7	0.9	1	VH	0.75	1	1	0.525	0.9	1
		MD9	VH	0.7	0.9	1	VH	0.75	1	1			
QFD6	Customization-new	MD10	VH	0.7	0.9	1	VH	0.75	1	1	0.525	0.9	1
		MD11	H	0.5	0.7	0.9	VH	0.75	1	1			
		MD12	VH	0.7	0.9	1	VH	0.75	1	1			
		MD13	VH	0.7	0.9	1	H	0.5	0.75	1			
QFD7	Time Behavior	MD14	H	0.5	0.7	0.9	VH	0.75	1	1	0.375	0.7	0.9
		MD15	H	0.5	0.7	0.9	H	0.5	0.75	1			
QFD8	Resource Behavior	MD16	VH	0.7	0.9	1	VH	0.75	1	1	0.525	0.9	1
		MD17	H	0.5	0.7	0.9	H	0.5	0.75	1			
		MD18	H	0.5	0.7	0.9	VH	0.75	1	1			
		MD19	H	0.5	0.7	0.9	VH	0.75	1	1			
		MD20	VH	0.7	0.9	1	H	0.5	0.75	1			
QFD9	Efficiency Compliance	MD21	H	0.5	0.7	0.9	H	0.5	0.75	1	0.25	0.525	0.9
QFD10	Scalability	MD22	H	0.5	0.7	0.9	H	0.5	0.75	1	0.375	0.7	0.9
		MD23	H	0.5	0.7	0.9	VH	0.75	1	1			
		MD24	H	0.5	0.7	0.9	VH	0.75	1	1			
QFD11	Performance	MD25	VH	0.7	0.9	1	VH	0.75	1	1	0.525	0.9	1
		MD26	VH	0.7	0.9	1	VH	0.75	1	1			
QFD12	Data-availability-new	MD27	H	0.5	0.7	0.9	VH	0.75	1	1	0.375	0.7	0.9
		MD28	H	0.5	0.7	0.9	VH	0.75	1	1			
		MD29	H	0.5	0.7	0.9	H	0.5	0.75	1			
QFD13	Replace-ability	MD30	VH	0.7	0.9	1	VH	0.75	1	1	0.525	0.9	1
QFD14	Adaptability	MD31	VH	0.7	0.9	1	VH	0.75	1	1	0.525	0.9	1
QFD15	Install-ability	MD32	VH	0.7	0.9	1	VH	0.75	1	1	0.525	0.9	1

QFD16	Co – existence	MD33	M	0.3	0.5	0.7	VH	0.75	1	1	0.225	0.5	0.7
QFD17	Portability Compliance	MD34	M	0.3	0.5	0.7	H	0.5	0.75	1	0.15	0.375	0.7
QFD18	Analyzability	MD35	H	0.5	0.7	0.9	H	0.5	0.75	1	0.525	0.9	1
		MD36	H	0.5	0.7	0.9	H	0.5	0.75	1			
		MD37	H	0.5	0.7	0.9	H	0.5	0.75	1			
		MD38	H	0.5	0.7	0.9	VH	0.75	1	1			
		MD39	VH	0.7	0.9	1	VH	0.75	1	1			
QFD19	Stability	MD40	VH	0.7	0.9	1	VH	0.75	1	1	0.525	0.9	1
		MD41	H	0.5	0.7	0.9	VH	0.75	1	1			
QFD20	Maintainability Compliance	MD42	H	0.5	0.7	0.9	H	0.5	0.75	1	0.25	0.525	0.9
QFD21	Fault Tolerance	MD43	VH	0.7	0.9	1	VH	0.75	1	1	0.525	0.9	1
		MD44	H	0.5	0.7	0.9	H	0.5	0.75	1			
QFD22	Recoverability	MD45	VH	0.7	0.9	1	H	0.5	0.75	1	0.35	0.675	1
QFD23	Reliability Compliance	MD46	H	0.5	0.7	0.9	H	0.5	0.75	1	0.25	0.525	0.9

Step2 Prioritization of fuzzy weight with pair wise comparison as per decision makers

This section demonstrate the relative ranking of fuzzy weight in between quality factors, where relative comparison among quality factors are decided by the decision makers of mobile company. This section implement algorithm as discussed in section 5.3.1, chapter 5. Pair wise comparison executed with the help of a matrix in between QFD1 to QFD23 as illustrated by table 6.2.

Table 6.2: Pair-wise Comparison Matrix among all QFs for Developer

	QFD1	*QFD2*	*QFD3*	*QFD4*	*QFD5*	*QFD6*	*QFD7*	*QFD8*	*QFD9*	*QFD10*	*QFD11*	*QFD12*	*QFD13*	*QFD14*	*QFD15*	*QFD16*	*QFD17*	*QFD18*	*QFD19*	*QFD20*	*QFD21*	*QFD22*	*QFD23*	*GFW*			*NFW*			***Crisp-wt***
QFD1	1	¼	1	1	1/7	1/5	1/3	2	1	1/3	1/7	1/8	1	1	1	½	1	1/3	1	1	1/5	¼	1	0.45	0.52	0.61	0.012375	0.01768	0.02562	0.019
QFD2	4	1	1	1	¼	½	1/3	2	1	1	1/5	1/3	½	1/3	1/3	½	1	¼	½	1	1/5	¼	1	0.48	0.58	0.76	0.0132	0.01972	0.03192	0.022
QFD3	1	1	1	1	1/5	3	1/3	¼	4	1	1/5	3	1	1	1	1	4	1/3	½	5	½	½	5	0.81	0.97	1.2	0.022275	0.03298	0.0504	0.035
QFD4	1	1	1	1	1/6	1/5	¼	¼	1	1/3	1/7	1/5	½	½	½	½	1	1/5	¼	1	1/3	¼	1	0.37	0.44	0.56	0.010175	0.01496	0.02352	0.016
QFD5	7	4	5	6	1	3	2	2	7	3	1/3	1	2	2	2	2	7	1	2	7	1	1	7	1.75	2.41	3	0.048125	0.08194	0.126	0.085
QFD6	5	2	1/3	5	1/3	1	½	½	7	1	1/3	½	1/3	1/3	1/3	1/3	7	¼	1/3	6	1/5	1/5	6	0.65	0.84	1.15	0.017875	0.02856	0.0483	0.032
QFD7	3	3	3	4	½	2	1	5	6	3	¼	1	2	2	2	2	6	2	2	5	1	1	6	1.46	2.09	2.71	0.04015	0.07106	0.11382	0.075
QFD8	½	½	4	4	½	2	1/5	1	5	3	1/5	2	1	1	1	1	5	1	1	5	½	½	5	1.02	1.24	1.62	0.02805	0.04216	0.06804	0.046
QFD9	1	1	¼	1	1/7	1/7	1/6	1/5	1	¼	1/5	1/5	1/5	1/5	1/5	1/5	1	¼	¼	1	1/6	1/6	1	0.29	0.33	0.38	0.007975	0.01122	0.01596	0.012
QFD10	3	1	1	3	1/3	1	1/3	1/3	4	1	¼	1	1	1	1	1	5	1	1	5	1	1	5	1.04	1.17	1.33	0.0286	0.03978	0.05586	0.041
QFD11	7	5	5	7	3	3	4	5	5	4	1	4	3	3	3	3	7	3	3	7	1	1	6	2.71	3.47	4.2	0.074525	0.11798	0.1764	0.123
QFD12	8	3	1/3	5	1	2	1	½	5	1	¼	1	½	½	½	½	3	¼	¼	4	1/5	1/5	3	0.73	0.95	1.29	0.020075	0.0323	0.05418	0.036
QFD13	1	2	1	2	½	3	½	1	5	1	1/3	2	1	1	1	1	4	1/5	1/5	3	¼	¼	3	0.78	0.99	1.25	0.02145	0.03366	0.0525	0.036
QFD14	1	3	1	2	½	3	½	1	5	1	1/3	2	1	1	1	1	3	1/5	1/5	4	1/3	1/3	4	0.83	1.06	1.32	0.022825	0.03604	0.05544	0.038
QFD15	1	3	1	2	½	3	½	1	5	1	1/3	2	1	1	1	1	3	¼	¼	4	1/3	1/3	3	0.78	1.16	1.34	0.02145	0.03944	0.05628	0.039
QFD16	2	2	1	2	½	3	½	1	5	1	1/3	2	1	1	1	1	3	¼	¼	4	1/3	1/3	3	0.81	1.07	1.39	0.022275	0.03638	0.05838	0.039
QFD17	1	1	¼	1	1/7	1/7	1/6	1/5	1	1/5	1/7	1/3	¼	1/3	1/3	1/3	1	1/6	1/6	1	1/6	1/6	½	0.28	0.33	0.4	0.0077	0.01122	0.0168	0.012
QFD18	3	4	3	5	1	4	½	1	4	1	1/3	4	5	5	4	4	6	1	1	5	1/3	1/3	4	1.6	2	2.47	0.044	0.068	0.10374	0.072
QFD19	1	2	2	4	½	3	½	1	4	1	1/3	4	5	5	4	4	6	1	1	4	1/3	1/3	4	1.32	1.71	2.2	0.0363	0.05814	0.0924	0.062
QFD20	1	1	1/5	1	1/7	1/6	1/5	1/5	1	1/5	1/7	¼	1/3	¼	¼	¼	1	1/5	¼	1	1/5	1/5	1	0.3	0.34	0.4	0.00825	0.01156	0.0168	0.012
QFD21	5	5	2	3	1	5	1	2	6	1	1	5	4	3	3	3	6	3	3	5	1	1/3	4	1.92	2.5	3.04	0.0528	0.085	0.12768	0.088
QFD22	4	4	2	4	1	5	1	2	6	1	1	5	4	3	3	3	6	3	3	5	3	1	5	2.11	2.75	3.34	0.058025	0.0935	0.14028	0.097
QFD23	1	1	1/5	1	1/7	1/6	1/6	1/5	1	1/5	1/6	1/3	1/3	1/4	1/3	1/3	2	1/4	1/4	1	1/4	1/5	1	0.31	0.37	0.46	0.008525	0.01258	0.01932	0.013

On the basis of crisp weight obtained for each quality factors, decision makers allocate five rating fuzzy weight criteria to each quality factor as illustrated in table 6.3.

Table 6.3: Allocation of Fuzzy Weight for Developer

Fuzzy Weight	Allocation as per crisp value of quality factors
VH	QFD5, QFD7, QFD11, QFD18, QFD19, QFD21, QFD22
H	QFD2, QFD3, QFD6, QFD8, QFD10, QFD12, QFD13, QFD14, QFD15, QFD16
M	QFD1, QFD4, QFD9, QFD17, QFD20, QFD23

Step3. Evaluation of overall fuzzy rating as per developer for each quality factor as illustrated in table 6.4.

Table 6.4: Evaluation of Overall Fuzzy Rate: Developer's View

	Quality Factors	Fuzzy rate			Fuzzy weight			Overall fuzzy rate(QF)		
QFD1	Suitability	0.525	0.9	1	0.25	0.5	0.75	0.13125	0.45	0.75
QFD2	Accuracy	0.525	0.9	1	0.5	0.75	1	0.2625	0.675	1
QFD3	Interoperability	0.525	0.9	1	0.5	0.75	1	0.2625	0.675	1
QFD4	Functionality Compliance	0.25	0.525	0.9	0.25	0.5	0.75	0.0625	0.2625	0.675
QFD5	Security	0.525	0.9	1	0.75	1	1	0.39375	0.9	1
QFD6	Customization-new	0.525	0.9	1	0.5	0.75	1	0.2625	0.675	1
QFD7	Time Behavior	0.375	0.7	0.9	0.75	1	1	0.28125	0.7	0.9
QFD8	Resource Behavior	0.525	0.9	1	0.5	0.75	1	0.2625	0.675	1
QFD9	Efficiency Compliance	0.25	0.525	0.9	0.25	0.5	0.75	0.0625	0.2625	0.675
QFD10	Scalability	0.375	0.7	0.9	0.5	0.75	1	0.1875	0.525	0.9
QFD11	Performance	0.525	0.9	1	0.75	1	1	0.39375	0.9	1
QFD12	Data-availability-new	0.375	0.7	0.9	0.5	0.75	1	0.1875	0.525	0.9

QFD13	Replace-ability	0.525	0.9	1	0.5	0.75	1	0.2625	0.675	1
QFD14	Adaptability	0.525	0.9	1	0.5	0.75	1	0.2625	0.675	1
QFD15	Install-ability	0.525	0.9	1	0.5	0.75	1	0.2625	0.675	1
QFD16	Co – existence	0.225	0.5	0.7	0.5	0.75	1	0.1125	0.375	0.7
QFD17	Portability Compliance	0.15	0.375	0.7	0.25	0.5	0.75	0.0375	0.1875	0.525
QFD18	Analyzability	0.525	0.9	1	0.75	1	1	0.39375	0.9	1
QFD19	Stability	0.525	0.9	1	0.75	1	1	0.39375	0.9	1
QFD20	Maintainability Compliance	0.25	0.525	0.9	0.25	0.5	0.75	0.0625	0.2625	0.675
QFD21	Fault Tolerance	0.525	0.9	1	0.75	1	1	0.39375	0.9	1
QFD22	Recoverability	0.35	0.675	1	0.75	1	1	0.2625	0.675	1
QFD23	Reliability Compliance	0.25	0.525	0.9	0.25	0.5	0.75	0.0625	0.2625	0.675

Step4. Evaluation of overall fuzzy rating as per developer view:

Assessment of overall fuzzy rating utilizing the equation outlined in section 4.5.1.

Overall fuzzy rating as per developer = (0.39375, 0.9, 1)

6.2.2 Quantification of quality factors using MAQM-MA framework :Tester's view

The tester has a total of 43 metrics, MT1 to MT43 and 26 quality factors, QFT1 to QFT26. Fuzzy weight and fuzzy rate are the two inputs that are used for each measure. This calculation adopts algorithm as discussed in sections 5.3.1 and 5.3.3 of chapter 5. Fuzzy rating for each quality factor was examined with the assistance of both fuzzy inputs that were given for metrics, and as a result, an overall fuzzy rating of the tester viewpoints was generated with the help of fuzzy weight and fuzzy rating of quality factors.

The following steps will illustrate the evaluation:

Step1. Evaluate overall quality (fuzzy rating) for quality factors QFT1 to QFT26 with the help of fuzzy weight and fuzzy rate assigned for each metrics as shown in Table 6.5. The questionnaire facilitates the assignment of fuzzy rates for the metrics (APPENDIX-2).

Table 6.5: Evaluation of Fuzzy Rating for each QF: Tester's View

SN	Quality Factors	fuzzy rate(metric)					fuzzy weight(metric)				Fuzzy Rating (Quality Factors)		
QFT1	Suitability	MT1	H	0.5	0.7	0.9	VH	0.75	1	1	0.375	0.7	0.9
QFT2	Accuracy	MT2	H	0.5	0.7	0.9	VH	0.75	1	1	0.375	0.7	0.9
		MT3	M	0.3	0.5	0.7	VH	0.75	1	1			
QFT3	Interoperability	MT4	H	0.5	0.7	0.9	VH	0.75	1	1	0.375	0.7	0.9
		MT5	H	0.5	0.7	0.9	H	0.5	0.75	1			
		MT6	M	0.3	0.5	0.7	VH	0.75	1	1			
QFT4	Functionality Compliance	MT7	M	0.3	0.5	0.7	VH	0.75	1	1	0.225	0.5	0.7
QFT5	Security	MT8	H	0.5	0.7	0.9	VH	0.75	1	1	0.375	0.7	0.9
QFT6	Customization-new	MT9	H	0.5	0.7	0.9	VH	0.75	1	1	0.375	0.7	0.9
		MT10	H	0.5	0.7	0.9	H	0.5	0.75	1			
		MT11	M	0.3	0.5	0.7	H	0.5	0.75	1			
		MT12	H	0.5	0.7	0.9	VH	0.75	1	1			
QFT7	Time Behavior	MT13	H	0.5	0.7	0.9	VH	0.75	1	1	0.375	0.7	0.9
QFT8	Resource Behavior	MT14	M	0.3	0.5	0.7	H	0.5	0.75	1	0.225	0.5	0.7
		MT15	M	0.3	0.5	0.7	VH	0.75	1	1			
QFT9	Efficiency	MT16	H	0.5	0.7	0.9	VH	0.75	1	1	0.375	0.7	0.9

	Compliance												
QFT10	scalability	MT17	M	0.3	0.5	0.7	H	0.5	0.75	1	0.225	0.5	0.7
		MT18	M	0.3	0.5	0.7	VH	0.75	1	1			
		MT19	M	0.3	0.5	0.7	VH	0.75	1	1			
QFT11	Performance	MT20	H	0.5	0.7	0.9	VH	0.75	1	1	0.375	0.7	0.9
		MT21	M	0.3	0.5	0.7	VH	0.75	1	1			
		MT22	M	0.3	0.5	0.7	H	0.5	0.75	1			
		MT23	M	0.3	0.5	0.7	H	0.5	0.75	1			
QFT12	Data-availability-new	MT24	M	0.3	0.5	0.7	VH	0.75	1	1	0.225	0.5	0.7
QFT13	Replace-ability	MT25	H	0.5	0.7	0.9	VH	0.75	1	1	0.375	0.7	0.9
QFT14	Adaptability	MT26	H	0.5	0.7	0.9	VH	0.75	1	1	0.375	0.7	0.9
QFT15	Install-ability	MT27	M	0.3	0.5	0.7	VH	0.75	1	1	0.225	0.5	0.7
QFT16	Co – existence	MT28	L	0.1	0.3	0.5	VH	0.75	1	1	0.075	0.3	0.5
QFT17	Portability Compliance	MT29	L	0.1	0.3	0.5	VH	0.75	1	1	0.075	0.3	0.5
QFT18	Analyzability	MT30	H	0.5	0.7	0.9	VH	0.75	1	1	0.375	0.7	0.9
QFT19	Stability	MT31	H	0.5	0.7	0.9	VH	0.75	1	1	0.375	0.7	0.9
QFT20	Maintainability Compliance	MT32	H	0.5	0.7	0.9	VH	0.75	1	1	0.375	0.7	0.9
QFT21	Fault Tolerance	MT33	M	0.3	0.5	0.7	VH	0.75	1	1	0.225	0.5	0.7
QFT22	Recoverability	MT34	M	0.3	0.5	0.7	VH	0.75	1	1	0.225	0.5	0.7
QFT23	Reliability Compliance	MT35	M	0.3	0.5	0.7	VH	0.75	1	1	0.225	0.5	0.7
QFT24	Changeability	MT36	M	0.3	0.5	0.7	H	0.5	0.75	1	0.225	0.5	0.7
		MT37	M	0.3	0.5	0.7	VH	0.75	1	1			
QFT25	Testability	MT38	M	0.3	0.5	0.7	H	0.5	0.75	1	0.375	0.7	0.9
		MT39	H	0.5	0.7	0.9	H	0.5	0.75	1			
		MT40	H	0.5	0.7	0.9	VH	0.75	1	1			

		MT41	M	0.3	0.5	0.7	H	0.5	0.75	1			
QFT26	Maturity	MT42	M	0.3	0.5	0.7	H	0.5	0.75	1	0.375	0.7	0.9
		MT43	H	0.5	0.7	0.9	VH	0.75	1	1			

Step2 Prioritization of fuzzy weight with pair wise comparison as per decision makers

This section demonstrate the relative ranking of fuzzy weight in between quality factors, where relative comparison among quality factors are decided by the decision makers of mobile company. This section implement algorithm as discussed in section5.3.1, chapter 5. Pair wise comparison executed with the help of a matrix in between QFT1 to QFT26 as illustrated by table 6.6.

Table 6.6: Pair-wise Comparison Matrix among all QFs for Tester

	QFT1	QFT2	QFT 3	QFT 4	QFT 5	QFT 6	QFT 7	QFT 8	QFT 9	QFT 10	QFT 11	QFT 12	QFT 13	QFT 14	QFT 15	QFT 16	QFT 17	QFT 18	QFT 19	QFT 20	QFT 21	QFT 22	QFT 23	QFT 24	QFT 25	QFT 26	GFW			NFW		Crisp-wt	
QFT1	1	¼	1	1	1/7	1/5	1/3	2	1	1/3	1/7	1/8	1	1	1	½	1	1/3	1	1	1/5	¼	1	½	1/3	3	0.465	0.543	0.67	0.011	0.017	0.026	0.018
QFT2	4	1	1	1	¼	½	1/3	2	1	1	1/5	1/3	½	1/3	1/3	½	1	¼	½	1	1/5	¼	1	3	1	5	0.562	0.688	0.893	0.014	0.021	0.035	0.023
QFT3	1	1	1	1	1/5	3	1/3	¼	4	1	1/5	3	1	1	1	1	4	1/3	½	5	½	½	5	1	1/3	3	0.83	0.97	1.193	0.02	0.03	0.047	0.032
QFT4	1	1	1	1	1/6	1/5	¼	¼	1	1/3	1/7	1/5	½	½	½	½	1	1/5	¼	1	1/3	¼	1	1/5	1/6	1/3	0.34	0.412	0.516	0.008	0.013	0.02	0.013
QFT5	7	4	5	6	1	3	2	2	7	3	1/3	1	2	2	2	2	7	1	2	7	1	1	7	5	½	6	1.755	2.409	3.051	0.043	0.074	0.119	0.078
QFT6	5	2	1/3	5	1/3	1	½	½	7	1	1/3	½	1/3	1/3	1/3	1/3	7	¼	1/3	6	1/5	1/5	6	1	4	4	0.741	0.955	1.277	0.018	0.029	0.05	0.032
QFT7	3	3	3	4	½	2	1	5	6	3	¼	1	2	2	2	2	6	2	2	5	1	1	6	1	1	4	1.456	2.006	2.544	0.036	0.062	0.1	0.066
QFT8	½	½	4	4	½	2	1/5	1	5	3	1/5	2	1	1	1	1	5	1	1	5	½	½	5	1	1	4	1.061	1.271	1.622	0.026	0.039	0.063	0.042
QFT9	1	1	¼	1	1/7	1/7	1/6	1/5	1	¼	1/5	1/5	1/5	1/5	1/5	1/5	1	¼	¼	1	1/6	1/6	1	1/5	1/5	¼	0.275	0.315	0.456	0.007	0.01	0.018	0.011
QFT10	3	1	1	3	1/3	1	1/3	1/3	4	1	¼	1	1	1	1	1	5	1	1	5	1	1	5	1	½	3	1.019	1.162	1.355	0.025	0.036	0.053	0.038
QFT11	7	5	5	7	3	3	4	5	5	4	1	4	3	3	3	3	7	3	3	7	1	1	6	5	1	5	2.664	3.379	4.071	0.066	0.104	0.159	0.109
QFT12	8	3	1/3	5	1	2	1	½	5	1	¼	1	½	½	½	½	3	¼	¼	4	1/5	1/5	3	3	1	4	0.813	1.056	1.4	0.02	0.033	0.055	0.036
QFT13	1	2	1	2	½	3	½	1	5	1	1/3	2	1	1	1	1	4	1/5	1/5	3	¼	¼	3	1	1	1	0.8	0.99	1.22	0.02	0.031	0.048	0.033
QFT14	1	3	1	2	½	3	½	1	5	1	1/3	2	1	1	1	1	3	1/5	1/5	4	1/3	1/3	4	3	1	1	0.867	1.092	1.344	0.021	0.034	0.053	0.036
QFT15	1	3	1	2	½	3	½	1	5	1	1/3	2	1	1	1	1	3	¼	¼	4	1/3	1/3	3	4	1	2	0.832	1.231	1.432	0.02	0.038	0.056	0.038
QFT16	2	2	1	2	½	3	½	1	5	1	1/3	2	1	1	1	1	3	¼	¼	4	1/3	1/3	3	1/3	1/3	1	0.747	0.965	1.273	0.018	0.03	0.05	0.032
QFT17	1	1	¼	1	1/7	1/7	1/6	1/5	1	1/5	1/7	1/3	¼	1/3	1/3	1/3	1	1/6	1/6	1	1/6	1/6	½	1/5	1/5	¼	0.267	0.315	0.387	0.007	0.01	0.015	0.01
QFT18	3	4	3	5	1	4	½	1	4	1	1/3	4	5	5	4	4	6	1	1	5	1/3	1/3	4	1	1	4	1.57	1.932	2.353	0.039	0.06	0.092	0.063
QFT19	1	2	2	4	½	3	½	1	4	1	1/3	4	5	5	4	4	6	1	1	4	1/3	1/3	4	1	½	4	1.28	1.632	2.12	0.032	0.05	0.083	0.055
QFT20	1	1	1/5	1	1/7	1/6	1/5	1/5	1	1/5	1/7	¼	1/3	¼	¼	¼	1	1/5	¼	1	1/5	1/5	1	¼	1/6	¼	0.275	0.324	0.387	0.007	0.01	0.015	0.01
QFT21	5	5	2	3	1	5	1	2	6	1	1	5	4	3	3	3	6	3	3	5	1	1/3	4	3	½	5	1.833	2.419	3.006	0.045	0.075	0.118	0.079
QFT22	4	4	2	4	1	5	1	2	6	1	1	5	4	3	3	3	6	3	3	5	3	1	5	5	½	5	2.046	2.684	3.324	0.05	0.083	0.13	0.087
QFT23	1	1	1/5	1	1/7	1/6	1/6	1/5	1	1/5	1/6	1/3	1/3	1/4	1/3	1/3	2	1/4	1/4	1	1/4	1/5	1	¼	1/6	¼	0.292	0.349	0.43	0.007	0.011	0.017	0.011
QFT24	2	1/3	1	5	1/5	1	1	1	5	1	1/5	1/3	1	1/3	1/4	3	5	1	1	4	1/3	1/5	4	1	1/5	1	0.79	0.94	1.1	0.019	0.029	0.043	0.03
QFT25	3	1	3	6	2	1/4	1	1	5	2	1	1	1	1	1	3	5	1	2	6	2	2	6	5	1	5	1.54	1.82	2.37	0.038	0.056	0.093	0.062
QFT26	1/3	1/5	1/3	3	1/6	1/4	1/4	1/4	4	1/3	1/5	1/4	1	1	1/2	1	4	1/4	1/4	4	1/5	1/5	4	1	1/5	1	0.44	0.54	0.81	0.011	0.017	0.032	0.02

On the basis of crisp weight obtained for each quality factors, decision makers allocate five rating fuzzy weight criteria to each quality factor as illustrated in table 6.7.

Table 6.7: Allocation of Fuzzy Weight for Tester

Fuzzy Weight	Allocation as per crisp vale of quality factors
VH	QFT5, QFT7, QFT8, QFT10, QFT11, QFT12, QFT14, QFT15, QFT18, QFT19, QFT21, QFT22, QFT25
H	QFT1, QFT2, QFT3, QFT6, QFT13, QFT16, QFT24, QFT26
M	QFT4, QFT9, QFT17, QFT20, QFT23

Step3. Evaluation of overall fuzzy rating as per tester for each quality factor as illustrated in table 6.8.

Table 6.8: Evaluation of Overall Fuzzy Rate: Tester's View

SN	Quality Factors	fuzzy rate(QF)				fuzzy weight(QF)				Overall fuzzy rating			
1	Suitability	VH	0.375	0.7	0.9	H	0.5	0.75	1	H1	0.1875	0.525	0.9
2	Accuracy	VH	0.375	0.7	0.9	H	0.5	0.75	1	H1	0.1875	0.525	0.9
3	Interoperability	VH	0.375	0.7	0.9	H	0.5	0.75	1	H1	0.1875	0.525	0.9
4	Functionality Compliance	H	0.225	0.5	0.7	M	0.25	0.5	0.75	L1	0.05625	0.25	0.525
5	Security	VH	0.375	0.7	0.9	VH	0.75	1	1	VH	0.28125	0.7	0.9
6	Customization-new	VH	0.375	0.7	0.9	H	0.5	0.75	1	H1	0.1875	0.525	0.9

7	Time Behaviour	VH	0.375	0.7	0.9	VH	0.75	1	1	VH	0.28125	0.7	0.9
8	Resource Behaviour	H	0.225	0.5	0.7	VH	0.75	1	1	H2	0.16875	0.5	0.7
9	Efficiency Compliance	VH	0.375	0.7	0.9	M	0.25	0.5	0.75	M2	0.09375	0.35	0.675
10	scalability	H	0.225	0.5	0.7	VH	0.75	1	1	H2	0.16875	0.5	0.7
11	Performance	VH	0.375	0.7	0.9	VH	0.75	1	1	VH	0.28125	0.7	0.9
12	Data-availability-new	H	0.225	0.5	0.7	VH	0.75	1	1	H2	0.16875	0.5	0.7
13	Replace-ability	VH	0.375	0.7	0.9	H	0.5	0.75	1	H1	0.1875	0.525	0.9
14	Adaptability	VH	0.375	0.7	0.9	VH	0.75	1	1	VH	0.28125	0.7	0.9
15	Install-ability	H	0.225	0.5	0.7	VH	0.75	1	1	H2	0.16875	0.5	0.7
16	Co – existence	M	0.075	0.3	0.5	H	0.5	0.75	1	L2	0.0375	0.225	0.5
17	Portability Compliance	M	0.075	0.3	0.5	M	0.25	0.5	0.75	VL	0.01875	0.15	0.375
18	Analysability	VH	0.375	0.7	0.9	VH	0.75	1	1	VH	0.28125	0.7	0.9
19	Stability	VH	0.375	0.7	0.9	VH	0.75	1	1	VH	0.28125	0.7	0.9
20	Maintainability Compliance	VH	0.375	0.7	0.9	M	0.25	0.5	0.75	M2	0.09375	0.35	0.675
21	Fault Tolerance	H	0.225	0.5	0.7	VH	0.75	1	1	H2	0.16875	0.5	0.7
22	Recoverability	H	0.225	0.5	0.7	VH	0.75	1	1	H2	0.16875	0.5	0.7

23	Reliability Compliance	H	0.225	0.5	0.7	M	0.25	0.5	0.75	L1	0.05625	0.25	0.525
24	Changeability	H	0.225	0.5	0.7	H	0.5	0.75	1	M1	0.1125	0.375	0.7
25	Testability	VH	0.375	0.7	0.9	VH	0.75	1	1	VH	0.28125	0.7	0.9
26	Maturity	VH	0.375	0.7	0.9	H	0.5	0.75	1	H1	0.1875	0.525	0.9

Step4. Evaluation of overall fuzzy rating as per tester view:

Assessment of overall fuzzy rating utilizing the equation outlined in section 4.5.1.

Overall fuzzy rating as per tester = (0.2813, 0.7, 0.9)

6.2.3 Quantification of quality factors using MAQM-MA framework:β-User's view

This section provides an analysis of the β-user's point of view. This section adopts quality standardISO/IEC-9241-11, the β-user has a total of 42 metrics and six quality factors. Fuzzy weight and fuzzy rate are the two inputs that are used for each measure. With the assistance of both fuzzy inputs that were allocated for metrics, the overall fuzzy rate of quality factor was analyzed, and as a result, an overall fuzzy rating of β-user was generated. This study adopts algorithm as discussed insections 5.3.1 and 5.3.3 of chapter 5, which evaluate appropriate fuzzy weight for quality factors as per decision makers.

The following steps will illustrate the evaluation:

Step1. Design a quality factor framework for M-Commerce application, illustrated as table 6.9.

Table 6.9: Quality Factors for β-User According toISO/IEC-9241-11

	Quality factors
QFβU1	features
QFβU2	time-taken
QFβU3	learnability
QFβU4	accuracy
QFβU5	security
QFβU6	feedback

Step2. Evaluate average fuzzy rating as per the opinion of 33 β-users for each metrics, illustrated as per table 6.10. The questionnaire facilitates the assignment of fuzzy rates for the metrics (APPENDIX-3).

Table 6.10: Evaluation of Average Fuzzy Rating According to 33 β-Users

QF		Metric	VH	H	M	L	VL	Average Fuzzy Rating		
features	MβU1	Consistency in text, font and colors	2	14	14	2	1	0.39	0.58	0.78
	MβU2	Easy to navigation in mobile app	4	12	14	1	2	0.4	0.59	0.78
	MβU3	Support night vision	3	10	10	7	3	0.33	0.52	0.71
	MβU4	Finding information about product are easier	3	13	13	3	1	0.39	0.58	0.78
	MβU5	Speech to text converted accurately	5	15	7	4	2	0.41	0.6	0.79
	MβU6	Description of each product is accurate	4	12	15	1	1	0.41	0.6	0.79

	MβU7	Product photograph displayed adequately	7	14	10	1	1	0.45	0.65	0.83
	MβU8	Prices of products are adequately shown	6	16	9	1	1	0.45	0.65	0.83
	MβU9	Status (available, out of stock) of each product is adequately shown	8	14	7	4	0	0.46	0.66	0.83
	MβU10	Easy to register	9	15	7	2	0	0.49	0.69	0.86
	MβU11	Easy to change customer information	9	15	8	0	1	0.49	0.69	0.86
	MβU12	Easy to order product	8	17	6	1	1	0.48	0.68	0.86
	MβU13	Shopping carts information is accurate	9	14	9	1	0	0.49	0.69	0.86
	MβU14	Adequate information about how to order	8	17	7	0	1	0.49	0.69	0.86
	MβU15	Adequate information about payment options	8	15	9	1	0	0.48	0.68	0.86
	MβU16	Adequate information about how to cancel the product	5	15	11	1	1	0.44	0.63	0.82
	MβU17	Adequate information about	8	15	9	1	0	0.48	0.68	0.86

		return & refund policy								
	MβU18	Adequate information about order detail	7	14	11	1	0	0.46	0.66	0.84
	MβU19	Adequate information about delivery time	3	13	14	3	0	0.4	0.6	0.79
	MβU20	Adequate information about delivery cost	4	17	8	2	2	0.42	0.62	0.8
	MβU21	Adequate information about delivery area	7	12	11	2	1	0.44	0.63	0.81
	MβU22	Delivery to other address	4	14	9	4	2	0.39	0.58	0.77
	MβU23	Online order tracking available	4	16	10	1	2	0.42	0.62	0.8
time-taken	MβU24	Search particular product	1	6	14	8	4	0.26	0.45	0.65
	MβU25	Time to complete a task	2	8	12	9	2	0.3	0.49	0.69
learbability	MβU26	Easy to learn interface	4	12	12	5	0	0.39	0.59	0.78
	MβU27	Adequate content management	3	15	11	3	1	0.4	0.6	0.79
	MβU28	Adequate help (demo version)	3	10	13	7	0	0.35	0.55	0.75
	MβU29	Adequate help (text version)	2	11	16	3	1	0.36	0.56	0.75

accuracy	MβU30	Mobile app respond properly as per action	4	12	15	2	0	0.41	0.61	0.8
	MβU31	Every component of interface respond accurately	7	10	14	1	1	0.43	0.63	0.81
	MβU32	Probability to search successfully task completion in 1st attempt	7	9	13	3	1	0.41	0.61	0.79
	MβU33	Probability to completion of task within given time	6	16	11	0	0	0.47	0.67	0.85
security	MβU34	Adequate information about privacy policy	5	15	11	1	1	0.44	0.63	0.82
	MβU35	Secure socket layer used by mobile app	4	20	7	2	0	0.46	0.66	0.85
	MβU36	Well recognized secure payment methods	5	16	11	0	1	0.45	0.65	0.83
	MβU37	Different mode for verification such as OTP based	7	15	11	0	0	0.48	0.68	0.85
feedback	MβU38	Overall features	3	14	14	2	0	0.41	0.61	0.8
	MβU39	Overall learning process	3	15	13	1	1	0.41	0.61	0.8
	MβU40	Overall accuracy	4	17	9	2	1	0.43	0.63	0.82
	MβU41	Overall security	5	19	9	0	0	0.48	0.68	0.86
	MβU42	Overall experience to use this mobile app	5	12	14	1	1	0.42	0.62	0.8

Step 3. Evaluate overall quality (fuzzy rating) for quality factors QFβU1 to QFβU6 with the help of fuzzy weight and fuzzy rate assigned for each metrics MβU1 to MβU42, illustrated as per Table 6.11 and Table 6.12.

Table 6.11: Fuzzy Rate and Fuzzy Weight for each Metrics: β-User's View

QF	Metric	Metric	Fuzzy rating			Fuzzy weight			
Features	MβU1	Consistency in text, font and colors	0.39	0.58	0.78	H	0.5	0.75	1
	MβU2	Easy to navigation in mobile app	0.4	0.59	0.78	VH	0.75	1	1
	MβU3	Support night vision	0.33	0.52	0.71	H	0.5	0.75	1
	MβU4	Finding information about product are easier	0.39	0.58	0.78	VH	0.75	1	1
	MβU5	Speech to text converted accurately	0.41	0.6	0.79	H	0.5	0.75	1
	MβU6	Description of each product is accurate	0.41	0.6	0.79	VH	0.75	1	1
	MβU7	Product photograph displayed adequately	0.45	0.65	0.83	H	0.5	0.75	1
	MβU8	Prices of products are adequately shown	0.45	0.65	0.83	H	0.5	0.75	1
	MβU9	Status (available, out of stock) of each product is adequately shown	0.46	0.66	0.83	H	0.5	0.75	1
	MβU10	Easy to register	0.49	0.69	0.86	VH	0.75	1	1
	MβU11	Easy to change customer information	0.49	0.69	0.86	H	0.5	0.75	1
	MβU12	Easy to order product	0.48	0.68	0.86	VH	0.75	1	1
	MβU13	Shopping cart's information is accurate	0.49	0.69	0.86	H	0.5	0.75	1
	MβU14	Adequate information about how to order	0.49	0.69	0.86	VH	0.75	1	1
	MβU15	Adequate information about payment options	0.48	0.68	0.86	VH	0.75	1	1

	MβU16	Adequate information about how to cancel the product	0.44	0.63	0.82	M	0.25	0.5	0.75
	MβU17	Adequate information about return & refund policy	0.48	0.68	0.86	M	0.25	0.5	0.75
	MβU18	Adequate information about order detail	0.46	0.66	0.84	H	0.5	0.75	1
	MβU19	Adequate information about delivery time	0.4	0.6	0.79	H	0.5	0.75	1
	MβU20	Adequate information about delivery cost	0.42	0.62	0.8	H	0.5	0.75	1
	MβU21	Adequate information about delivery area	0.44	0.63	0.81	VH	0.75	1	1
	MβU22	Delivery to other address	0.39	0.58	0.77	H	0.5	0.75	1
	MβU23	Online order tracking available	0.42	0.62	0.8	VH	0.75	1	1
Time taken	MβU24	Search particular product	0.26	0.45	0.65	VH	0.75	1	1
	MβU25	Time to complete a task	0.3	0.49	0.69	H	0.5	0.75	1
Learn-ability	MβU26	Easy to learn interface	0.39	0.59	0.78	VH	0.75	1	1
	MβU27	Adequate content management	0.4	0.6	0.79	H	0.5	0.75	1
	MβU28	Adequate help (demo version)	0.35	0.55	0.75	VH	0.75	1	1
	MβU29	Adequate help (text version)	0.36	0.56	0.75	H	0.5	0.75	1
Accuracy	MβU30	Mobile app respond properly as per action	0.41	0.61	0.8	H	0.5	0.75	1
	MβU31	Every component of interface respond accurately	0.43	0.63	0.81	VH	0.75	1	1
	MβU32	Probability to search successfully task completion in 1st attempt	0.41	0.61	0.79	VH	0.75	1	1
	MβU33	Probability to completion of task within given time	0.47	0.67	0.85	H	0.5	0.75	1

Security	MβU34	Adequate information about privacy policy	0.44	0.63	0.82	M	0.25	0.5	0.75
	MβU35	Secure socket layer used by mobile app	0.46	0.66	0.85	VH	0.75	1	1
	MβU36	Well recognized secure payment methods	0.45	0.65	0.83	VH	0.75	1	1
	MβU37	Different mode for verification such as OTP based	0.48	0.68	0.85	VH	0.75	1	1
Feedback	MβU38	Overall features	0.41	0.61	0.8	H	0.5	0.75	1
	MβU39	Overall learning process	0.41	0.61	0.8	H	0.5	0.75	1
	MβU40	Overall accuracy	0.43	0.63	0.82	VH	0.75	1	1
	MβU41	Overall security	0.48	0.68	0.86	VH	0.75	1	1
	MβU42	Overall experience to use this mobile app	0.42	0.62	0.8	VH	0.75	1	1

Table 6.12: Evaluation of Overall Fuzzy Rating for each Quality Factors: β-User's View

	Quality Factors	Fuzzy Rating (QF)		
QFβU1	features	0.3675	0.69	0.86
QFβU2	time-taken	0.195	0.45	0.69
QFβU3	learnability	0.2925	0.59	0.79
QFβU4	accuracy	0.3225	0.63	0.85
QFβU5	security	0.36	0.68	0.85
QFβU6	feedback	0.36	0.68	0.86

Step 4. Detailed evaluation process to prioritize fuzzy weight of quality factors.

The following steps are outlined, which provides an explanation of the approach that is used to determine the relative importance of all quality elements by using pair-wise comparison as per FW-MA algorithm discussed in section 5.3.1 .

(a) Design a pair-wise comparison framework matrix for each pair of quality factors as per the criteria imposed upon quality factors, illustrated as per table 6.13.

Decision makers are mutually agreed upon fallowing criteria imposed upon six quality factors:

1. features is 5 times more important than learnability.
2. features is 4 times more important than feedback.
3. time-taken is 5 times more important than learnability.
4. time-taken is 3 times more important than feedback.
5. learnability is 4 times more important than feedback.
6. accuracy is 6 times more important than features.
7. accuracy is 7 times more important than time-taken.
8. accuracy is 7 times more important than learnability.
9. accuracy is 5 times more important than feedback.
10. security is 6 times more important than features.
11. security is 8 times more important than time-taken.
12. security is 8 times more important than learnability.
13. security is 2 times more important than accuracy.
14. security is 5 times more important than feedback.

Table 6.13: Pair-wise Comparison Matrix among all Quality Factors for β-User

	QFβU1	QFβU2	QFβU3	QFβU4	QFβU5	QFβU6
QFβU1	1	1	5	1/6	1/6	4
QFβU2	1	1	5	1/7	1/8	3
QFβU3	1/5	1/5	1	1/7	1/8	4
QFβU4	6	7	7	1	½	5
QFβU5	6	8	8	2	1	5

QFβU6	¼	1/3	¼	1/5	1/5	1

(b) Replace crisp value by fuzzy value as per table 6.14 and table 6.15 and calculate fuzzy geometric mean (FGM) for each quality factor (QF), illustrated as per table 6.16.

Table 6.14:Linguistic Variable and Corresponding Fuzzy Value

Comparison (crisp)	Linguistic variable	Fuzzy value
1	Equally important	(1,1,1)
2	In between equally important and weakly importance	(1,2,3)
3	Weakly importance	(2,3,4)
4	In between weakly important and strongly importance	(3,4,5)
5	Strongly importance	(4,5,6)
6	In between strongly important and very strongly importance	(5,6,7)
7	Very strongly important	(6,7,8)
8	In between very strongly important and absolutely importance	(7,8,9)
9	Absolutely important	(8,9,9)

Table 6.15:Crisp Value and Corresponding Inverse Fuzzy Value

Comparison (crisp)	Fuzzy value (Inverse)	Simplified fuzzy Value
1/1	$(1,1,1)^{-1}$	(1,1,1)
1/2	$(1,2,3)^{-1}$	$\left(\frac{1}{3},\frac{1}{2},\frac{1}{1}\right)$
1/3	$(2,3,4)^{-1}$	$\left(\frac{1}{4},\frac{1}{3},\frac{1}{2}\right)$
1/4	$(3,4,5)^{-1}$	$\left(\frac{1}{5},\frac{1}{4},\frac{1}{3}\right)$
1/5	$(4,5,6)^{-1}$	$\left(\frac{1}{6},\frac{1}{5},\frac{1}{4}\right)$
1/6	$(6,7,8)^{-1}$	$\left(\frac{1}{7},\frac{1}{6},\frac{1}{5}\right)$
1/7	$(6,7,8)^{-1}$	$\left(\frac{1}{8},\frac{1}{7},\frac{1}{6}\right)$
1/8	$(7,8,9)^{-1}$	$\left(\frac{1}{9},\frac{1}{8},\frac{1}{7}\right)$
1/9	$(8,9,9)^{-1}$	$\left(\frac{1}{9},\frac{1}{9},\frac{1}{8}\right)$

Computation of FGM for quality factor QFβU1:

$((1*1*4*1/7*1/7*3)^{1/6},(1*1*5*1/6*1/6*4)^{1/6},(1*1*6*1/5*1/5*5)^{1/6})=(0.79, 0.91, 1.04)$

Similarly calculated for other quality factors.

Table 6.16: FGMComputation for each QF:β-Users

	QFβU1	QFβU2	QFβU3	QFβU4	QFβU5	QFβU6	FGM
QFβU1	(1,1,1)	(1,1,1)	(4,5,6)	(1/7,1/6,1/5)	(1/7,1/6,1/5)	(3,4,5)	(0.79,0.91,1.04)
QFβU2	(1,1,1)	(1,1,1)	(4,5,6)	(1/8,1/7,1/6)	(1/9,1/8,1/7)	(2,3,4)	(0.69,0.79,0.91)
QFβU3	(1/6,1/5,1/4)	(1/6,1/5,1/4)	(1,1,1)	(1/8,1/7,1/6)	(1/9,1/8,1/7)	(3,4,5)	(0.32,0.37,0.43)
QFβU4	(5,6,7)	(6,7,8)	(6,7,8)	(1,1,1)	(1/3,1/2,1)	(4,5,6)	(2.54,3.07,3.83)
QFβU5	(5,6,7)	(7,8,9)	(7,8,9)	(1,2,3)	(1,1,1)	(4,5,6)	(3.22,4.07,4.80)
QFβU6	(1/5,1/4,1/3)	(1/4,1/3,1/2)	(1/5,1/4,1/3)	(1/6,1/5,1/4)	(1/6,1/5,1/4)	(1,1,1)	(0.25,0.29,0.38)

(c) Computation of normalized fuzzy weight for each quality factors, illustrated astable 6.17.

Computation of normalized fuzzy weight for quality factor QFβU1:

$(0.79, 0.91, 1.04) * (\frac{1}{11.39}, \frac{1}{9.5}, \frac{1}{7.81}) = (0.069,0.096,0.133)$

Similarly calculated for other quality factors.

Table 6.17: NFW Computation for each QF:β-Users

	FGM	NFW
QFβU1	(0.79,0.91,1.04)	(0.069,0.096,0.133)
QFβU2	(0.69,0.79,0.91)	(0.061,0.083,0.117)
QFβU3	(0.32,0.37,0.43)	(0.028,0.039,0.055)
QFβU4	(2.54,3.07,3.83)	(0.223,0.323,0.490)
QFβU5	(3.22,4.07,4.80)	(0.283,0.428,0.615)
QFβU6	(0.25,0.29,0.38)	(0.022,0.031,0.049)

(d) Computation of crisp weight and priorities for each quality factors, illustrated as table 6.18.

Table 6.18: Priorities among QF's According to Crisp Weight: β-Users

	NFW	Crisp Weight	priority
QFβU1	(0.069,0.096,0.133)	0.098	3rd
QFβU2	(0.061,0.083,0.117)	0.087	4th
QFβU3	(0.028,0.039,0.055)	0.038	5th
QFβU4	(0.223,0.323,0.490)	0.345	2nd
QFβU5	(0.283,0.428,0.615)	0.442	1st
QFβU6	(0.022,0.031,0.049)	0.034	6th

(e) Computation of relative distance with in quality factors QFβU1 to QFβU6 as per crisp weight, illustrated as table 6.19.

Table 6.19: Relative Distance WithinQFs: β-Users

	Relative distance
QFβU5- QFβU4	0.097
QFβU4- QFβU1	0.247
QFβU1- QFβU2	0.011
QFβU2- QFβU3	0.049
QFβU3- QFβU6	0.004

(f) Computation of fuzzy weight on the basis of relative distance in between quality factors QFβU1 to QFβU5, decision makers decides the fuzzy weight. One possible combination, illustrated as table 6.20.

Table 6.20: Allocation of Appropriate Fuzzy Weight for β-Users

Quality factors	Fuzzy weight
QFβU5	VH
QFβU4	VH
QFβU1	H
QFβU2	H
QFβU3	M
QFβU6	M

Step 5. Evaluation of overall fuzzy rating as per developer for each quality factor as illustrated in table 6.21.

Table 6.21: Evaluation of Overall Fuzzy Rate:β-User'sView

QF	Fuzzy Rate				Fuzzy weight				Over-all Fuzzy Rate(β-user)		
QFβU1	VH	0.3675	0.69	0.86	H	0.5	0.75	1	0.18375	0.5175	0.86
QFβU2	VH	0.195	0.45	0.69	H	0.5	0.75	1	0.0975	0.3375	0.69
QFβU3	VH	0.2925	0.59	0.79	M	0.25	0.5	0.75	0.073125	0.295	0.5925
QFβU4	H	0.3225	0.63	0.85	VH	0.75	1	1	0.241875	0.63	0.85
QFβU5	VH	0.36	0.68	0.85	VH	0.75	1	1	0.27	0.68	0.85
QFβU6	VH	0.36	0.68	0.86	M	0.25	0.5	0.75	0.09	0.34	0.645

Step 6. Overall quality (fuzzy rating) as per β-Users evaluated as:

Overall quality = (0.3675,0.69,0.86)*(0.5,0.75,1) + (0.195,0.45,0.69)*(0.5,0.75,1) + (0.2925,0.59,0.79)*(0.25,0.5,0.75) + (0.3225,0.63,0.85)*(0.75,1,1) + (0.36,0.68,0.85)*(0.75,1,1) +(0.36,0.68,0.86)*(0.25,0.5,0.75)

Overall quality = (0.1838,0.5175,0.86) + (0.0975,0.3375,0.69) + (0.0731,0.295,0.5925) + (0.2419,0.63,0.85) + (0.27,0.68,0.85) + (0.09,0.34,0.645)

Overall quality = (max(0.1838,0.0975,0.0731,0.2419,0.27,0.09), max(0.5175,0.3375,0.295,0.63,0.68,0.34) , max(0.86,0.69,0.5925,0.85,0.85,0.645))

Overall quality (fuzzy rating) = (0.27, 0.68, 0.86)

6.3 QUANTIFICATION OF NET SOFTWARE QUALITY

This section evaluates overall quality as per developer, tester and β-user's perspectives. The fuzzy rating for each perspective given as:

Developer's view: (0.39, 0.9, 1)

Tester's view: (0.28, 0.7, 0.9)

β-user's view: (0.27,0.68,0.86)

The fuzzy weight for each perspective as per the decision maker of mobile application development firm given as:

Developer's view: (0.50, 0.75, 1)

Tester's view: (0.50, 0.75, 1)

β-user's view: (0.75, 1, 1)

Overall fuzzy quality rating for developer, tester and β-user view computed as:

= (0.39, 0.9, 1)* (0.50, 0.75, 1)+ (0.28, 0.7, 0.9)* (0.50, 0.75, 1)+ (0.27, 0.68, 0.86)* (0.75, 1, 1)

= (0.195, 0.675, 1) + (0.14, 0.525, 0.9) + (0.2025, 0.68, 0.86)

= (max (0.195, 0.14, 0.2025), max (0.675, 0.525, 0.68), max (1, 0.9,0.86))

= (0.2025, 0.68,1)

6.4 DEFUZZIFICATION

Defuzzification is the process which converts fuzzy rating in to crisp rating. This section presents defuzzification process for developer, tester and β-user's perspectives.

6.4.1 Crisp value: Developer's view

Fuzzy rating for developer's perspective: (0.39, 0.9, 1)

Computation of crisprating as per developer's view (onapplying centroid formula) as per Fig. 6.1:

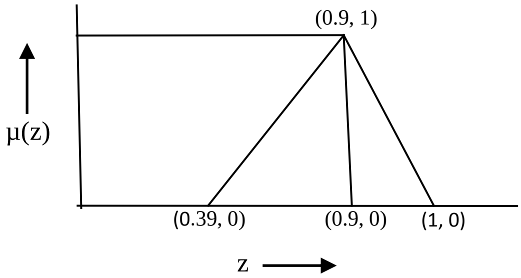


Fig 6.1: Defuzzificationfor Developer's View

Equationoflinepassingthrough(0.39,0)and(0.9,1):

$\mu\ (z) - 0 = \frac{1-0}{0.9-0.39}\ (z- 0.39)$

$\mu\ (z) = \frac{1}{0.51}\ (z- 0.39)$

$\mu\ (z) = 1.96\ z- 0.76$

Equation of line passing through (1, 0) and (0.9, 1):

$\mu\ (z) - 0 = \frac{1-0}{0.9-1.0}\ (z- 1.0)$

$\mu\ (z) = \frac{1}{-0.1}\ (z- 1.0)$

$\mu\ (z) = 10 – 10\ z$

$$Z^* = \frac{\int_{0.39}^{0.9}(1.96\ z - 0.76)\ z\ dz + \int_{0.9}^{1}(10 - 10\ z) z\ dz}{\int_{0.39}^{0.9}(1.96\ z - 0.76)\ dz + \int_{0.9}^{1}(10 - 10\ z)\ dz}$$

$Z^* = \frac{0.186 + 0.047}{0.257 + 0.05}$

$Z^* = \frac{0.233}{0.307}$

Z^*= 0.7589 or 75.89%

Thus the crisp value as per developer's perspective is 75.89%

6.4.2 Crisp value: Tester's view

Fuzzy rating for developer's perspective: (0.28, 0.7, 0.9)

Computation of crisp rating as per Tester's view (on applying centroid formula) as per Fig. 6.2:

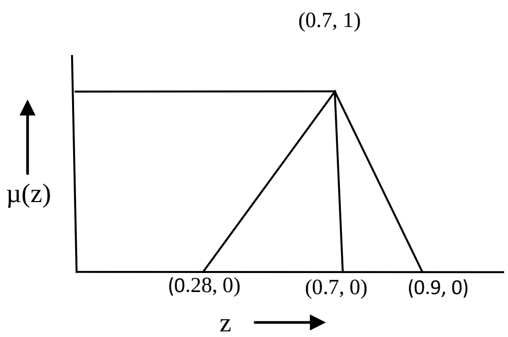


Fig 6.2: Defuzzificationfor Tester's View

Equation of line passing through (0.281, 0) and (0.7, 1):

$$\mu\,(z) - 0 = \frac{1-0}{0.7-0.281}\,(z - 0.281)$$

$$\mu\,(z) = \frac{1}{0.419}\,(z - 0.281)$$

$$\mu\,(z) = 2.39\,z - 0.671$$

Equation of line passing through (0.9, 0) and (0.7, 1):

$$\mu\,(z) - 0 = \frac{1-0}{0.7-0.9}\,(z - 0.9)$$

$$\mu\,(z) = \frac{1}{-0.2}\,(z - 0.9)$$

$$\mu\,(z) = 4.5 - 5\,z$$

$$Z^* = \frac{\int_{0.281}^{0.7}(2.39\,z - 0.671)\,z\,dz + \int_{0.7}^{0.9}(4.5 - 5\,z)z\,dz}{\int_{0.281}^{0.7}(2.39\,z - 0.671)\,dz + \int_{0.7}^{0.9}(4.5 - 5\,z)\,dz}$$

$$Z^* = \frac{0.1177 + 0.07667}{0.2095 + 0.1}$$

$$Z^* = \frac{0.19437}{0.3095}$$

Z^*= 0.628013 or 62.80%

Thus the crisp value as per tester's perspective is 62.80%

6.4.3 Crisp value:β-User's view

Fuzzy rating for β-user's perspective: (0.27, 0.68, 0.86)

Computation of crisp rating as per β-User's view (on applying centroid formula) as per Fig. 6.3:

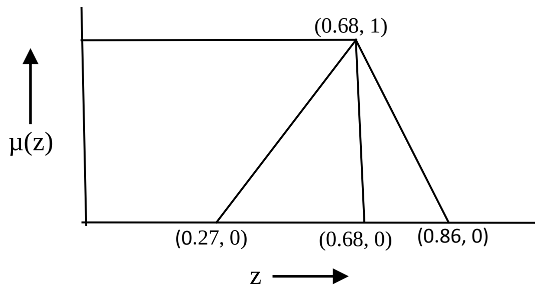


Fig 6.3: Defuzzificationfor β-User's View

Equation of line passing through (0.27, 0) and (0.68, 1):

$$\mu\,(z) - 0 = \frac{1-0}{0.68-0.27}\,(z - 0.27)$$

$$\mu\,(z) = \frac{1}{0.41}\,(z - 0.27)$$

$$\mu\,(z) = 2.44\,z - 0.66$$

Equation of line passing through (0.86, 0) and (0.68, 1):

$$\mu\,(z) - 0 = \frac{1-0}{0.68-0.86}\,(z - 0.86)$$

$$\mu\,(z) = \frac{1}{-0.18}\,(z - 0.86)$$

$$\mu\,(z) = 4.78 - 5.56\,z$$

$$Z^* = \frac{\int_{0.27}^{0.68}(2.44\,z - 0.66)\,z\,dz + \int_{0.68}^{0.86}(4.78 - 5.56\,z)z\,dz}{\int_{0.27}^{0.68}(2.44\,z - 0.66)\,dz + \int_{0.68}^{0.86}(4.78 - 5.56\,z)dz}$$

$$Z^* = \frac{0.111 + 0.0663}{0.205 + 0.09}$$

$$Z^* = \frac{0.1773}{0.295}$$

Z^*= 0.58746 or 58.75%

Thus the crisp value as per β-user's perspective is 58.75%

6.4.4 Overall crisp value (Net software quality)

Overall fuzzy rating as per three perspectives: (0.20, 0.68, 1.0)

Computation of crisp rating as per developer, tester and β-user's view (on applying centroid formula) as per Fig. 6.4:

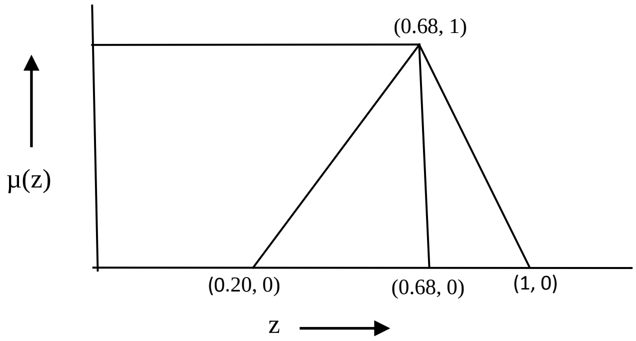


Fig 6.4: Defuzzificationfor Overall View

Equation of line passing through (0.20, 0) and (0.68, 1):

$\mu(z) - 0 = \frac{1-0}{0.68-0.20}(z - 0.20)$

$\mu(z) = \frac{1}{0.48}(z - 0.20)$

$\mu(z) = 2.083z - 0.417$

Equation of line passing through (1, 0) and (0.68, 1):

$\mu(z) - 0 = \frac{1-0}{0.68-1.0}(z - 1.0)$

$\mu(z) = \frac{1}{-0.32}(z - 1.0)$

$\mu(z) = 3.125 - 3.125z$

$$Z^* = \frac{\int_{0.20}^{0.68}(2.083z - 0.417)\,z\,dz + \int_{0.68}^{1}(3.125 - 3.125z)z\,dz}{\int_{0.20}^{0.68}(2.083z - 0.417)\,dz + \int_{0.68}^{1}(3.125 - 3.125z)\,dz}$$

$Z^* = \frac{0.1247+0.1259}{0.24+0.16}$

$Z^* = \frac{0.2506}{0.40}$

$Z^* = 0.6265$ or 62.65%

Thus the crisp value as per developer, tester and β-user's perspective is 62.65%

6.5SUMMARY

A fuzzy-based mathematical framework MAQM-MA, which converts a qualitative aspects in to quantitative assessment presented in this chapter with the help of a case study for a mobile commerce application. This framework assesses three aspects: the perspective of developer, tester, and β-user. There are two quality standards that are included into the MAQM-MA framework: ISO/IEC-9126 and ISO/IEC-9241-11. The development of an algorithm FW-MA that analyzes the importance of quality factors with the assistance of pair-wise comparisons for the developer, tester, and β-user is also shown in this chapter. Quality for three perspectives obtained as 77.39%, 62.80% and 58.75%. The overall quality of M-Commerce application obtained as 62.65%. Throughout the process of developing mobile applications, this quantification assists in gaining a better understanding of the overall quality and may take corrective measures if it's required.

CHAPTER 7

IMPLEMENTATION OF MAQM-MA FRAMEWORK

7.1 INTRODUCTION

This chapter presents a proposed software module that incorporates the fuzzy-based mathematical framework MAQM-MA for which the framework was proposed and discussed in chapter five and quantification of MAQM-MA in chapter six. This proposed software module has been developed using Java Script and Node.js. The developer view, the tester view, and the β-user view are the three perspectives that are considered while evaluating the software modules. The software module as per MAQM-MA framework combines four sub-modules. First three sub-modules represents developer, tester and β-user perspectives, while forth sub-module evaluates over all fuzzy rating based on three different views. The fuzzy rating will be defuzzified, which evaluate overall quality of any domain specific mobile application under development.

The overall quality depends upon two inputs: Fuzzy rate and fuzzy weight which have different roles. The role of fuzzy rate behaves like feedback of the developer, tester or β-user as per the prototype of mobile application under development. While fuzzy weight is according to the decision maker of mobile company, fuzzy weight will be allocated to quality factor as per the given priority of quality factor.

This chapter divided in to eight sections 7.1 to 7.8. Section 7.1 introduces about software implementation of fuzzy based mathematical frameworkMAQM-MA. Sections 7.2 explain about hardware and software configuration for software which implementsMAQM-MA through four sub-modules.Section 7.3 describes MAQM-MA as a software tool. Section 7.4 to section 7.7 explains about process of evaluation of overall fuzzy and crisp quality for developer, tester, β-user and their combined effect via sub-module-1 to sub-module-4. Section 7.8 concludes with summary of this chapter.

7.2 HARDWARE AND SOFTWARE CONFIGURATION

7.2.1 Hardware Requirement

The details of hardware requirements are as follows:

- Windows Edition: Windows 10 Pro
- Processor: Intel i5, 11th generation
- Installed RAM: 8GB
- Secondary Memory: 256 SSD / HDD
- System type: 64-bit operating system

7.2.2 Software Requirement

The details of software requirement are as follows [76]:

- Java Script
- Node.js (Run time)

7.3 DESCRIPTION OF MAQM-MA AS SOFTWARE TOOL

This software tool has four sub-modules that facilitate the implementation of the suggested fuzzy-based mathematical framework MAQM-MA. An evaluation of the overall fuzzy and crisp quality is performed by first three sub-modules, i.e. submodule-1, submodule-2 and submodule-3 for the developer, tester, and β-user. The total quality grade that is ultimately determined is dependent on the three views and evaluated by forth sub-module submodule-4.

7.4 DEVELOPER'S VIEW: SUBMODULE-1

This section implements the first sub-module of MAQM-MA, which evaluates overall fuzzy rating and then defuzzify to obtain overall crisp quality as per developer view. There are 46 metrics MD1 to MD46 and 23 quality factors QFD1 to QFD23 provided by the decision makers of mobile application development Company. Overall fuzzy rating and crisp quality evaluated as per the algorithm discussed in sections 5.3.1 and 5.3.3 chapter 5.

7.4.1 Allocation of fuzzy rate and weight for metrics

This section discusses the inputs provided to MAQM-MA as per developer. There are two inputs fuzzy rate and fuzzy weight. The user interface which accepts fuzzy rate and fuzzy weight for the metrics MD1 to MD46. Decision makers will provide fuzzy weight for each metrics, while developer allocate fuzzy rate for each metrics as per their feedback of prototype for mobile application under development. The submodule-1 of MAQM-MA will provide drop-down facility for five scale input as fuzzy rate and fuzzy weight in the form of five scale criteria very-high (VH), high (H), medium (M), low (L) and very-low (VL) as illustrated in fig 7.1.

Fig 7.1: Submodule-1: Fuzzy Rate and Weight for MD1 to MD46

7.4.2 Allocation of fuzzy rate and fuzzy weight for quality factor

Developer will evaluate fuzzy rating for each quality factor with the help of fuzzzy rate and fuzzy weight allocated to each metrics MD1 to MD46, now the next step of computation will be evaluation of fuzzy rating for each quality factors as illustrated in fig 7.2.

Fig 7.2: Sub-Module-1: Fuzzy Rate and Weight for QFD1 to QFD23

7.4.3 Evaluation of fuzzy rate and over all crisp quality

Now developer will evaluate final triangular fuzzy number or fuzzy rating as (0.39,0.9,1) and then defuzzified. The overall crisp quality will be obtained by defuzzification is 75.89 %, as illustrated in fig 7.3.

Final Dev Value

(0.39375, 0.9, 1)

Integrated final value

0.7589

Fig 7.3:Sub-Module-1: Overall Fuzzy and Crisp Quality for Developer

7.5 TESTER'S VIEW: SUBMODULE-2

This section implements the second sub-module of MAQM-MA, which evaluates overall fuzzy rating and then defuzzify to obtain overall crisp quality as per tester view. There are 43 metrics MT1 to MT43 and 26 quality factors QFT1 to QFT26 provided by the decision makers of mobile application development Company. Overall fuzzy rating and crisp quality evaluated as per the algorithm discussed in sections 5.3.1 and 5.3.3 chapter 5.

7.5.1 Allocation of fuzzy rate and fuzzy weight for metrics

This section discusses the inputs provided to MAQM-MA as per tester. There are two inputs fuzzy rate and fuzzy weight. The user interface which accepts fuzzy rate and fuzzy weight for the metrics MT1 to MT43. Decision makers will provide fuzzy weight for each metrics, while tester allocate fuzzy rate for each metrics as per their feedback of prototype for mobile application under development. The submodule-2 of MAQM-MA will provide drop-down facility for five scale input as fuzzy rate and fuzzy weight in the form of five scale criteria very-high (VH), high (H), medium (M), low (L) and very-low (VL) as illustrated in fig 7.4.

Fig 7.4:Sub-Module-2: Fuzzy Rate and Weight for MT1 to MT43

7.5.2 Allocation of fuzzy rate and fuzzy weight for quality factor

Tester will evaluate fuzzy rating for each quality factor with the help of fuzzy rate and fuzzy weight allocated to each metrics MT1 to MT43, now the next step of computation will be the evaluation of fuzzy rating for each quality factors as illustrated in Fig 7.5.

Fig 7.5Sub-Module-2: Fuzzy Rate and Weight for QFT1 to QFT26

7.5.3 Evaluation of fuzzy rate and over all crisp quality

Now tester will evaluate final triangular fuzzy number or fuzzy rating as (0.28,0.7,0.9) and then defuzzified. The overall crisp quality will be obtained by defuzzification is 62.80 %, as illustrated in fig 7.6.

Final Tester Value

(0.2813, 0.7, 0.9)

Integrated final value

0.62801300

Fig 7.6:Sub-Module-2: Overall Fuzzy and Crisp Quality for Tester

7.6 β-USERS’S VIEW: SUBMODULE-3

This section implements the third sub-module of MAQM-MA, which evaluates overall fuzzy rating and then defuzzify to obtain overall crisp quality as per β-Users. There are 42 metrics MβU1 to MβU42and 6 quality factors QFβU1 to QFβU6 provided by the decision makers of mobile application development Company. Overall fuzzy rating and crisp quality evaluated as per the algorithm discussed in sections 5.3.1 and 5.3.3 chapter 5.

7.6.1 Allocation of fuzzy rate and fuzzy weight for metrics

This section discusses the inputs provided to MAQM-MA as per β-User. There are two inputs fuzzy rate and fuzzy weight. The user interface which accepts fuzzy rate and fuzzy weight for the metrics MβU1 to MβU42. Decision makers will provide fuzzy weight for each metrics, while β-User allocate fuzzy rate for each metrics as per their feedback of prototype for mobile application under development. The submodule-3 of MAQM-MA will provide drop-down facility for five scale input as fuzzy weight in the form of five scale criteria very-high (VH), high (H), medium (M), low (L) and very-low (VL) as illustrated in Fig 7.7.

Fig 7.7Sub-Module-3: Fuzzy Rate and Weight for MβU1 to MβU42

7.6.2 Allocation of fuzzy rate and fuzzy weight for quality factor

β-Userswill evaluate fuzzy rating for each quality factor with the help of fuzzzy rate and fuzzy weight allocated to each metrics, now the next step of computation will be evaluation of fuzzy rating for each quality factorsas per β-User .

Fig 7.8:Sub-Module-3: Fuzzy Rate and Weight for QFβU1 to QFβU6

7.6.3 Evaluation of fuzzy rate and over all crisp quality

Now β-Users will evaluate final triangular fuzzy number or fuzzy rating as (0.27,0.68,0.86) and then defuzzified. The overall crisp quality will be obtained by defuzzification is 58.74 %, as illustrated in fig 7.8.

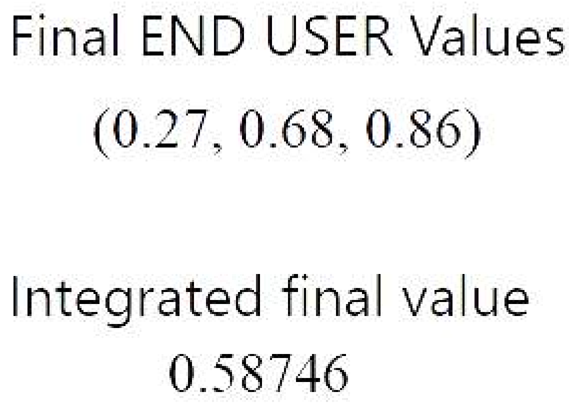


Fig 7.9:Sub-Module-3: Overall Fuzzy and Crisp Quality for β-User

7.7 OVERALL FUZZY AND CRISP RATING: SUBMODULE-4

This section implements the forth sub-module of MAQM-MA, which evaluates overall quality of a mobile application as per three perspectives, with the help of fuzzy rating evaluated in previous modules and fuzzy weight suggested by decision makers as illustrated in fig 7.9.

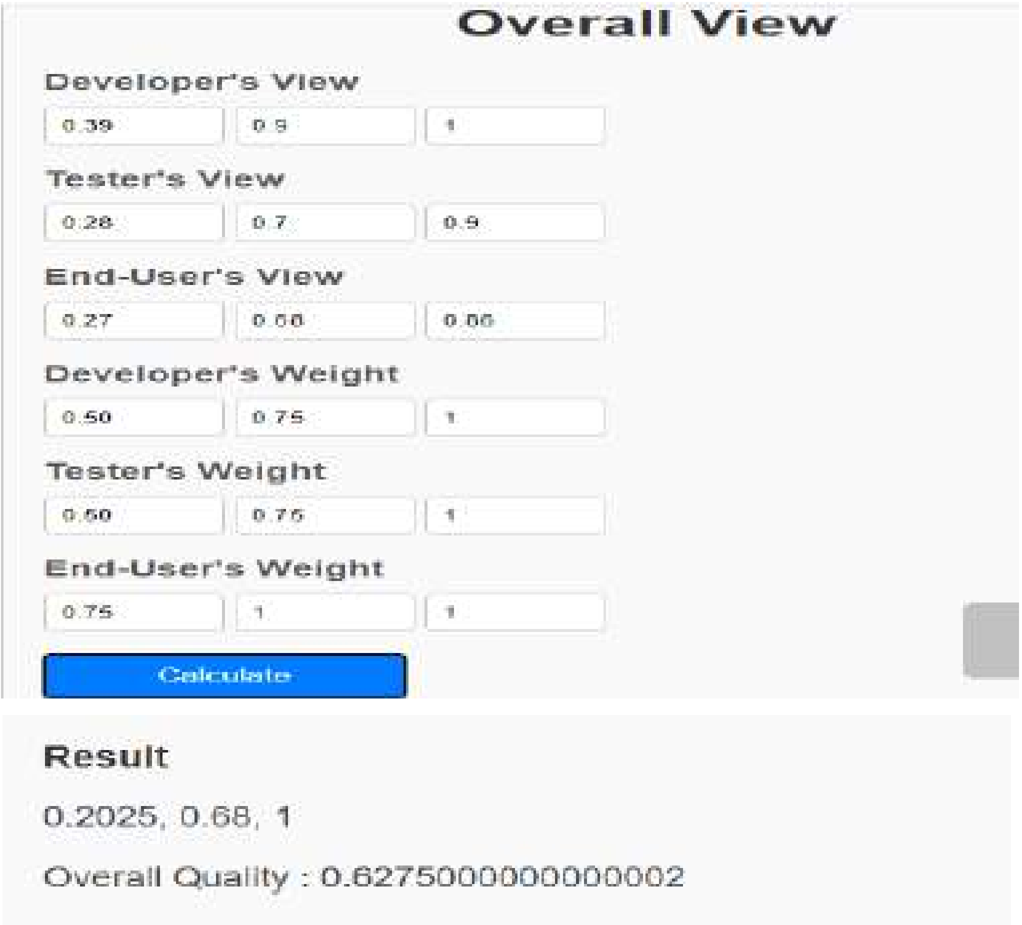


Fig 7.10:Sub-Module-4: Over-all Evaluation of Fuzzy Rating and Crisp Rating

7.8 SUMMARY

This chapter presents a software implementation of MAQM-MA framework, which has four sub-modules that facilitate the implementation of the suggested fuzzy-based mathematical framework MAQM-MA. An evaluation of the overall fuzzy and crisp quality is performed by first three sub-modules for the developer, tester, and β-user. The total quality grade that is ultimately determined is dependent on the three views and thus evaluated by forth sub-module of the software. This software module will access the overall quality of a mobile application during its development.

CHAPTER 8

VALIDATION OF MAQM-MA

8.1 INTRODUCTION

This section provides validation utilizing the MATLAB Fuzzy Logic Tool [75] from three perspectives: the developer, the tester, and the β-user,along with their combined effect. The overall quality as a results obtained in chapter six, simulated through fuzzy logic tool and simulated results shows the validation of proposed framework MAQM-MA.

This chapter is divided in to six sections 8.1 to 8.6. Section 8.1 focused on the process of validating data received by quantification using MAQM-MA framework, by the MATLAB Fuzzy Logic Tool. Section 8.2 explains about validation of MAQM-MA with the help of MATLAB Fuzzy Logic Tool as per perspective of developer. Section 8.3 explains about validation of MAQM-MA with the help of MATLAB Fuzzy Logic Tool as per perspective of tester. Section 8.4 explains about validation of MAQM-MA with the help of MATLAB Fuzzy Logic Tool as per perspective of β-User's. Section 8.5 explains about validation of MAQM-MA with the help of MATLAB Fuzzy Logic Tool as the combined effect of all three perspectives. All of the sections 8.2 to 8.5 are discussed with the help of fuzzy inference system (FIS), detail of fuzzy weight and fuzzy rate as inputs, over-all-rating as output and overall quality obtained numerically validated through control surface. Section 8.6 concludes with summary of this chapter.

8.2 VALIDATION: DEVELOPER'S VIEW

The following sub-sections will illustrate the procedure of quantitatively validating the overall quality of a mobile application using MATLAB Fuzzy Logic Tool, based on findings from chapter six.

8.2.1 Fuzzy Inference System (FIS) for Developer's View

This section explains about the implementation of FIS for developer's view with the help of Mamdani Type-1. The FIS includes two fuzzy inputs, Fuzzy-Weight and Fuzzy-Rate and Over-All-Rating-Developer as output along with fuzzy inference rules as illustrated in Fig.8.1. The property editor explains the property about FIS as illustrated in Fig 8.2 along with sample fuzzy rules as illustrated in Fig 8.3.

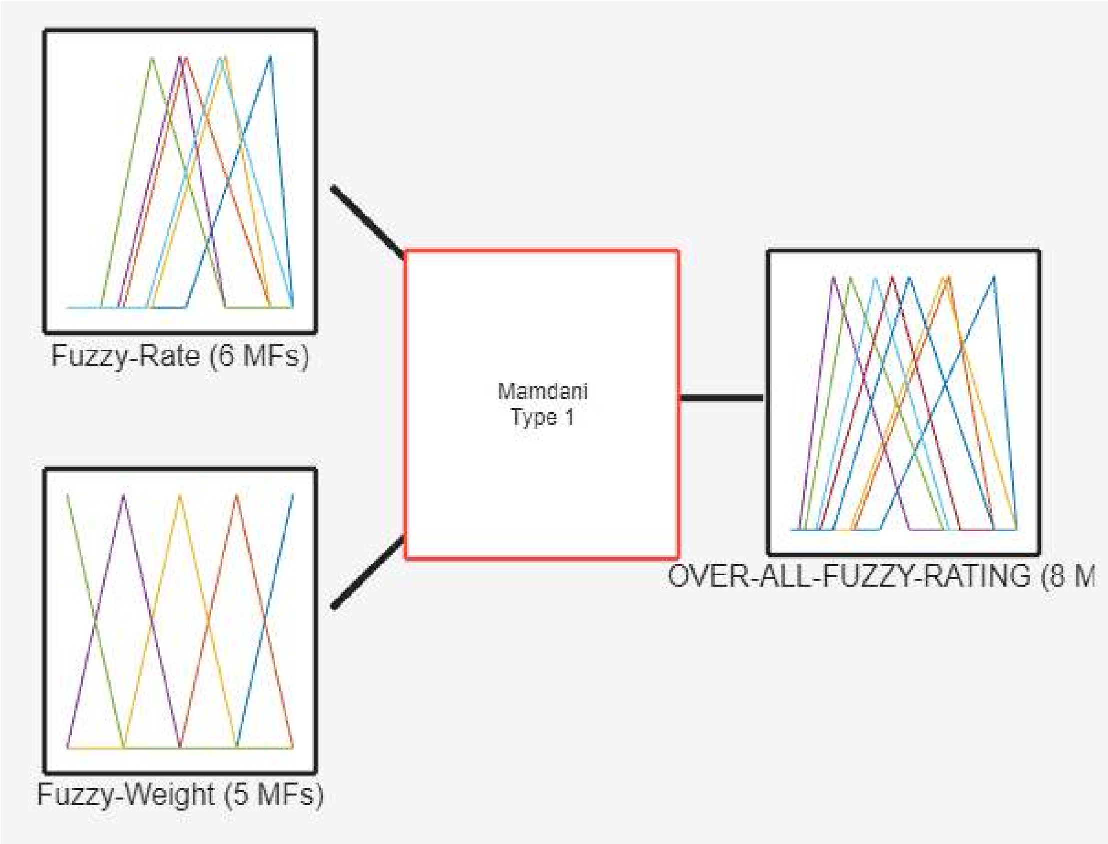


Fig 8.1:FIS: Developer's View

PROPERTY EDITOR: FIS

Type:	Mamdani Type-1
Name	Developer
And method	prod
Or method	max
Implication method	min
Aggregation method	max
Defuzzification method	centroid
Inputs:	2
Outputs:	1
Rules:	8

Fig 8.2:Property Editor (FIS): Developer's View

	Rule	Weight	Name
1	If Fuzzy-Rate is VH and Fuzzy-Weight is M then OVER-ALL-FUZZY-RATI...	1	rule1
2	If Fuzzy-Rate is VH and Fuzzy-Weight is H then OVER-ALL-FUZZY-RATIN...	1	rule2
3	If Fuzzy-Rate is VH and Fuzzy-Weight is VH then OVER-ALL-FUZZY-RATI...	1	rule3
4	If Fuzzy-Rate is H and Fuzzy-Weight is VH then OVER-ALL-FUZZY-RATIN...	1	rule4
5	If Fuzzy-Rate is L2 and Fuzzy-Weight is M then OVER-ALL-FUZZY-RATIN...	1	rule5
6	If Fuzzy-Rate is H and Fuzzy-Weight is H then OVER-ALL-FUZZY-RATIN...	1	rule6
7	If Fuzzy-Rate is L1 and Fuzzy-Weight is H then OVER-ALL-FUZZY-RATIN...	1	rule7
8	If Fuzzy-Rate is VL and Fuzzy-Weight is M then OVER-ALL-FUZZY-RATIN...	1	rule8

Fig 8.3:Sample Fuzzy Rules: Developer's View

8.2.2 Detail of Input Fuzzy-Weight

The input Fuzzy-Weight includes five triangular membership functions very-high (VH) to very-low (VL) illustrated in Fig. 8.4 along with the parameters illustrated in Fig. 8.5.

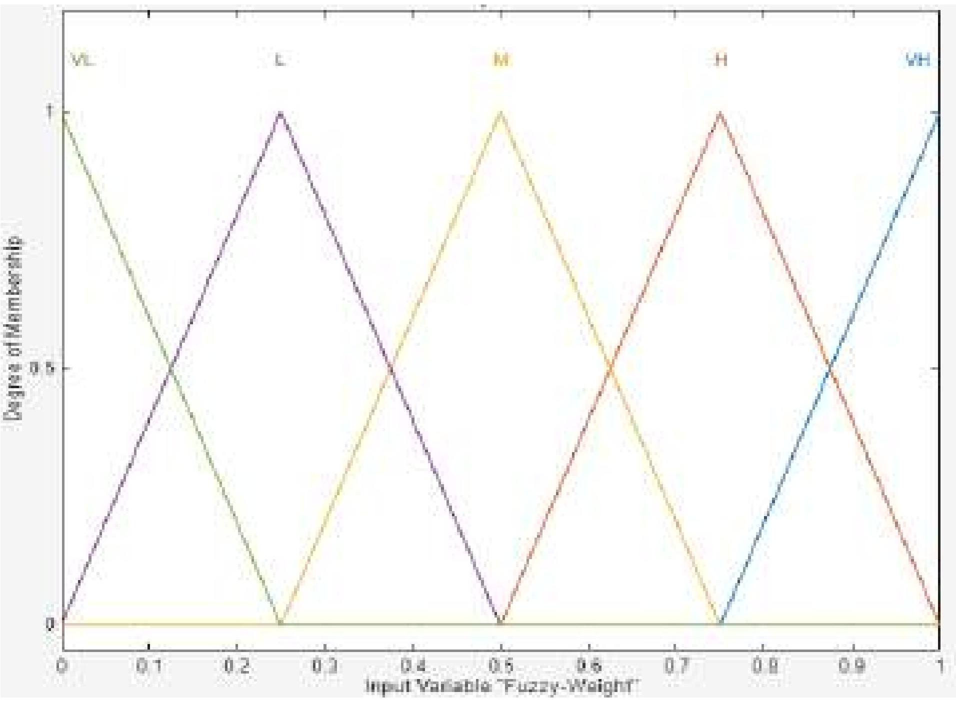
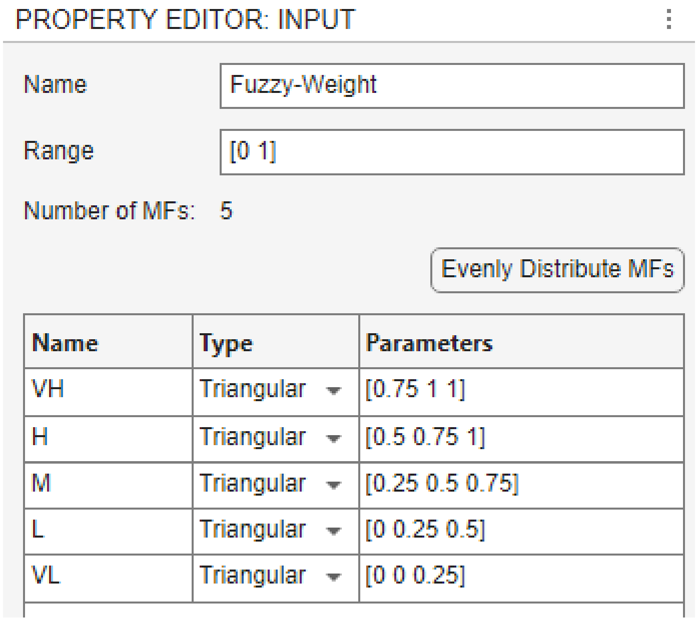


Fig 8.4: Membership Function Plot for Input (Fuzzy-Weight): Developer's View



Name	Type	Parameters
VH	Triangular	[0.75 1 1]
H	Triangular	[0.5 0.75 1]
M	Triangular	[0.25 0.5 0.75]
L	Triangular	[0 0.25 0.5]
VL	Triangular	[0 0 0.25]

Fig 8.5:Parameter Detail for Input (Fuzzy-Weight): Developer's View

8.2.3 Detail of Input Fuzzy-Rating

The input Fuzzy-Rating includes six triangular membership functions very-high (VH) to very-low (VL) illustrated in Fig.8.6 along with the parameters illustrated in Fig.8.7.

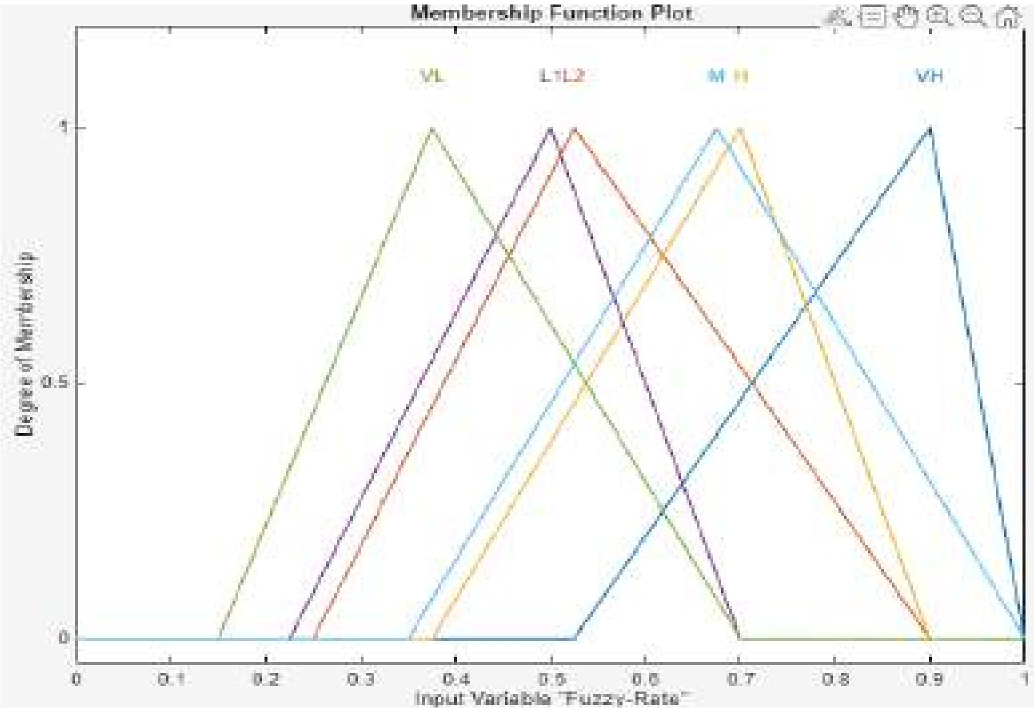


Fig 8.6:Membership Function Plot for Input (Fuzzy-Rating): Developer's View

PROPERTY EDITOR: INPUT

Name: Fuzzy-Rate

Range: [0 1]

Number of MFs: 6

Evenly Distribute MFs

Name	Type	Parameters
VH	Triangular	[0.525 0.9 1]
L2	Triangular	[0.25 0.525 0.9]
H	Triangular	[0.375 0.7 0.9]
L1	Triangular	[0.225 0.5 0.7]
VL	Triangular	[0.15 0.375 0.7]
M	Triangular	[0.35 0.675 1]

Fig 8.7:Parameter Detail for Input (Fuzzy-Rating): Developer's View

8.2.4 Detail of Output Over-All-Rating: Developer's View

The output Over-All-Fuzzy-Rating-Developer includes eight triangular membership functions, very high (VH) to very low (VL) illustrated in Fig.8.8 along with the parameters illustrated in Fig.8.9.

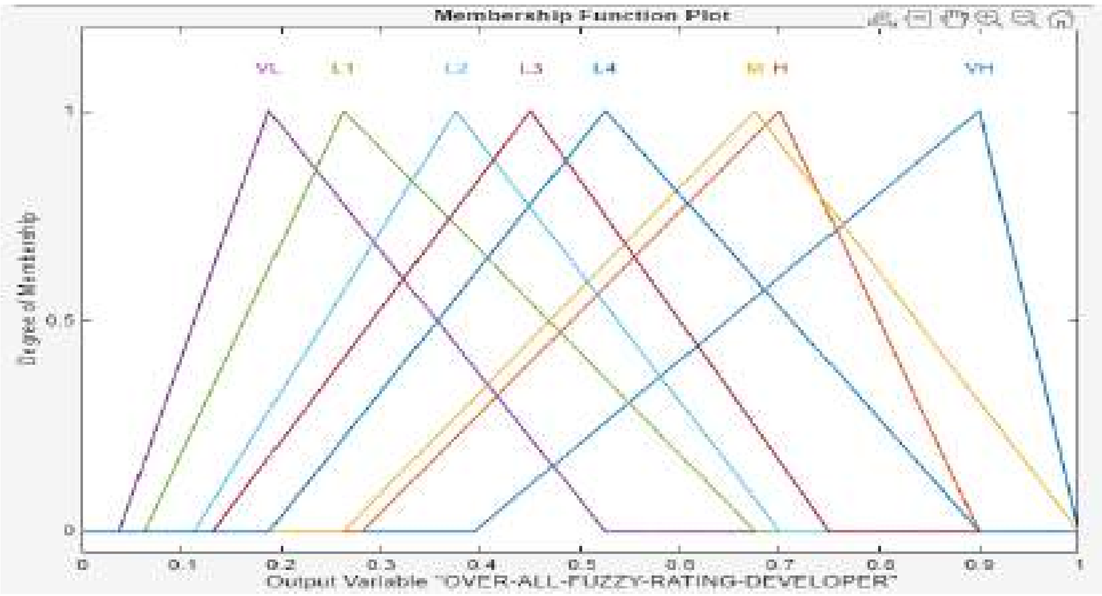


Fig 8.8:Membership Function Plot for Output (Over-All-Fuzzy-Rating): Developer's View

PROPERTY EDITOR: OUTPUT

Name OVER-ALL-FUZZY-RATING-DEVELOPEF

Range [0 1]

Number of MFs: 8

Evenly Distribute MFs

Name	Type	Parameters
VH	Triangular	[0.39375 0.9 1]
H	Triangular	[0.28125 0.7 0.9]
M	Triangular	[0.2625 0.675 1]
VL	Triangular	[0.0375 0.1875 0.525]
L1	Triangular	[0.0625 0.2625 0.675]
L2	Triangular	[0.1125 0.375 0.7]
L3	Triangular	[0.13125 0.45 0.75]
L4	Triangular	[0.1875 0.525 0.9]

Fig 8.9:Parameter Detail for Output (Over-All-Fuzzy-Rating): Developer's View

8.2.5 Control Surface for Over All Crisp Rating as per Developer

The following figure Fig.8.10, presents a numerical representation of the overall quality of the mobile application prototype currently being developed, based on the perspective of the developer.

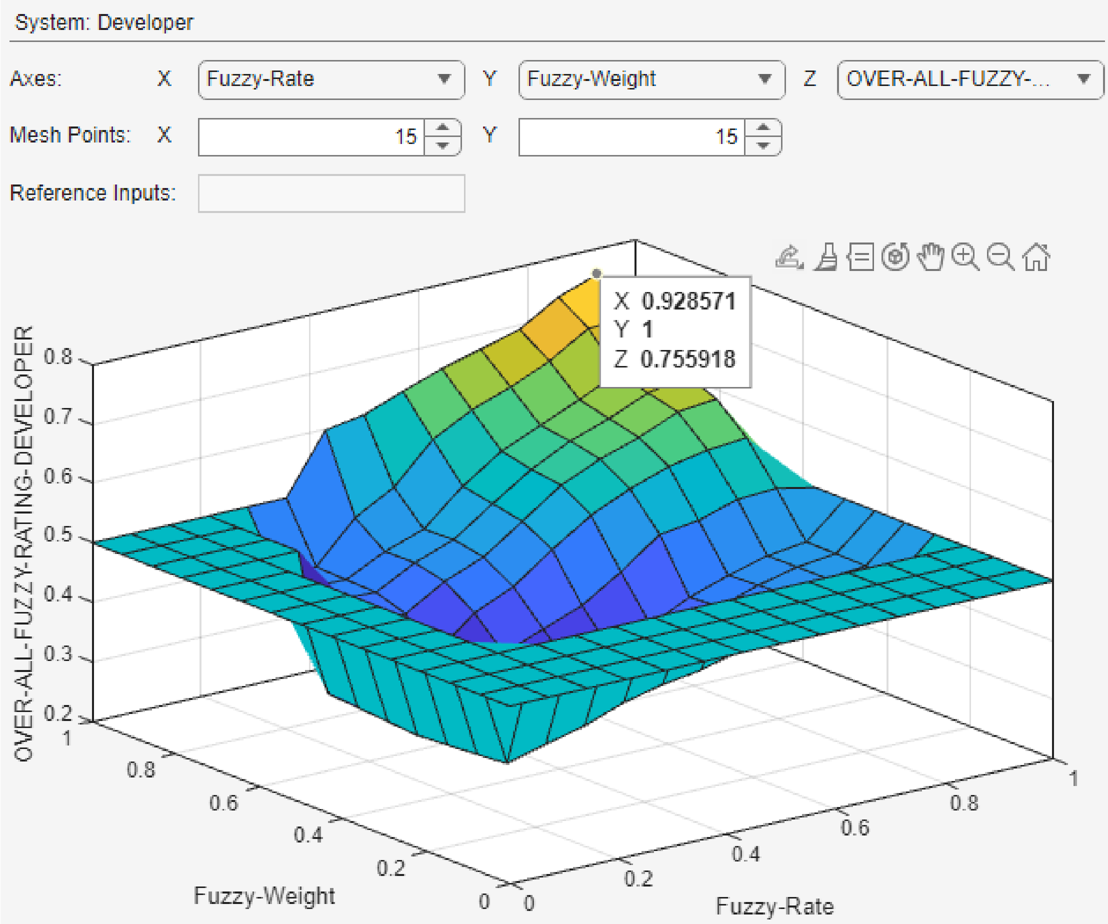


Fig 8.10:Control Surface for Over-All-Rating: Developer's View

The altitude Z is represented by the color of the surface in MATLAB surface plot (control surface), where X, Y and Z are illustrated as shown in Fig 8.10. The color changes as a result of change in the altitude.The overall quality of developer's view is 0.7589 or 75.89% as per proposed fuzzy based mathematical framework MAQM-MA. The overall quality obtained as per MATLAB simulation is 0.755918 or 75.59%. Thus the result obtained as per MAQM-MA verified.

8.3 VALIDATION: TESTER'S VIEW

The following sub-sections will illustrate the procedure of quantitatively validating the overall quality of a mobile application using MATLAB Fuzzy Logic Tool, based on findings from chapter six.

8.3.1 Fuzzy Inference System (FIS) for Tester's View

This section explains about the implementation of FIS for tester's view with the help of Mamdani Type-1. The FIS includes two fuzzy inputs, Fuzzy-Weight and Fuzzy-Rating and Over-All-Rating-Tester as output along with fuzzy inference rules as illustrated in Fig.8.11. The property editor explains the property about FIS as illustrated in Fig 8.12 along with sample fuzzy rules as illustrated in Fig 8.13.

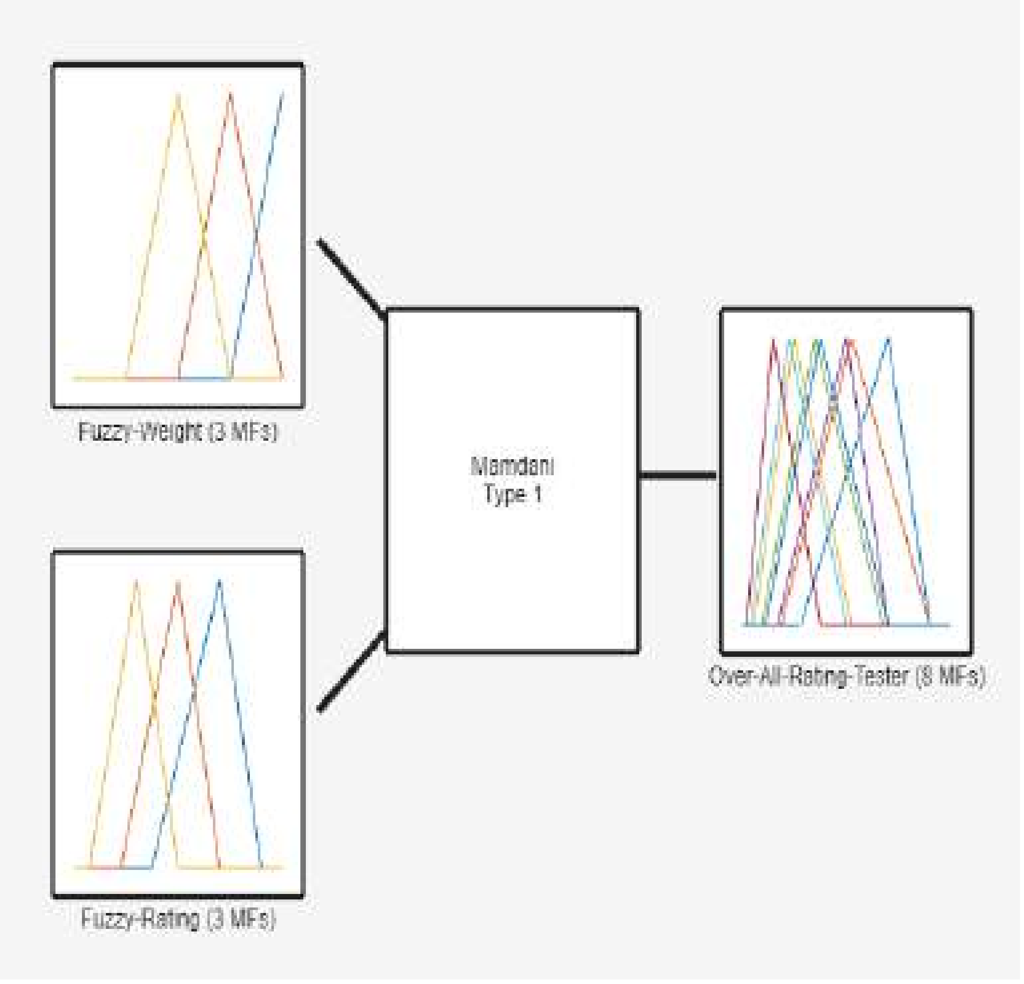


Fig 8.11:FIS: Tester's View

PROPERTY EDITOR: FIS

Type:	Mamdani Type-1
Name	Tester
And method	prod
Or method	max
Implication method	min
Aggregation method	max
Defuzzification method	centroid
Inputs:	2
Outputs:	1
Rules:	8

Fig 8.12:Property Editor (FIS): Tester's View

	Rule	Weight	Name
1	If Fuzzy-Weight is H and Fuzzy-Rating is VH then Over-All-Rating-Tester is...	1	rule1
2	If Fuzzy-Weight is M and Fuzzy-Rating is H then Over-All-Rating-Tester is L1	1	rule2
3	If Fuzzy-Weight is VH and Fuzzy-Rating is VH then Over-All-Rating-Tester ...	1	rule3
4	If Fuzzy-Weight is VH and Fuzzy-Rating is H then Over-All-Rating-Tester is...	1	rule4
5	If Fuzzy-Weight is M and Fuzzy-Rating is VH then Over-All-Rating-Tester i...	1	rule5
6	If Fuzzy-Weight is H and Fuzzy-Rating is M then Over-All-Rating-Tester is L2	1	rule6
7	If Fuzzy-Weight is M and Fuzzy-Rating is M then Over-All-Rating-Tester is VL	1	rule7
8	If Fuzzy-Weight is H and Fuzzy-Rating is H then Over-All-Rating-Tester is M1	1	rule8

Fig 8.13:Sample Fuzzy Rules: Tester's View

8.3.2 Detail of Input Fuzzy-Weight

The input Fuzzy-Weight includes three triangular membership functions very-high (VH), high (H) and medium (M)illustrated in Fig. 8.14 along with the parameters illustrated in Fig. 8.15.

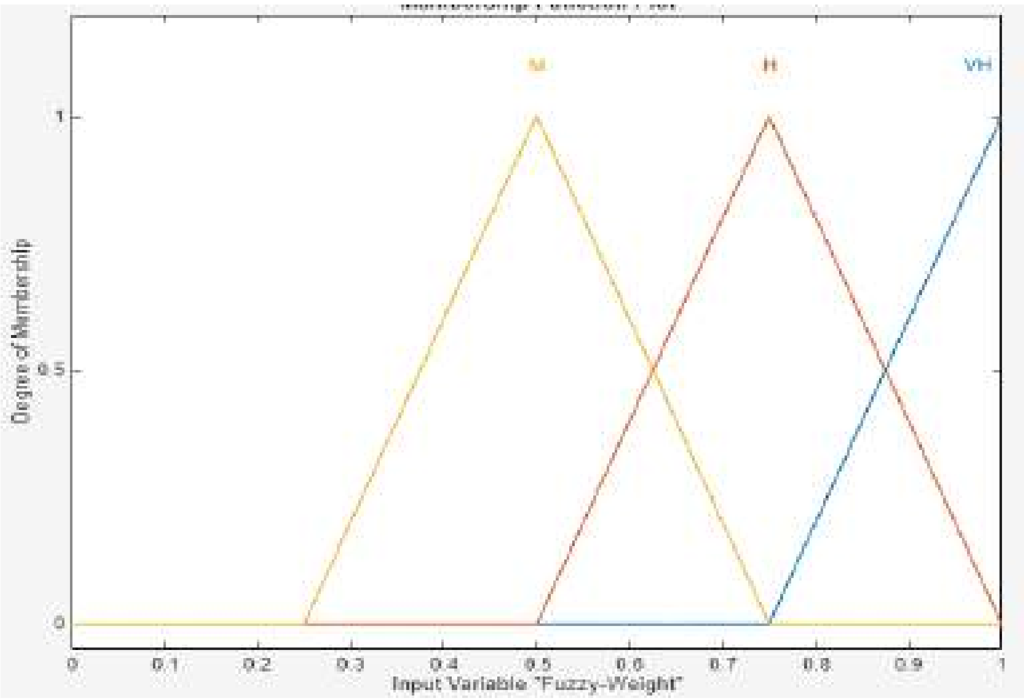


Fig 8.14:Membership Function Plot for Input (Fuzzy-Weight): Tester's View

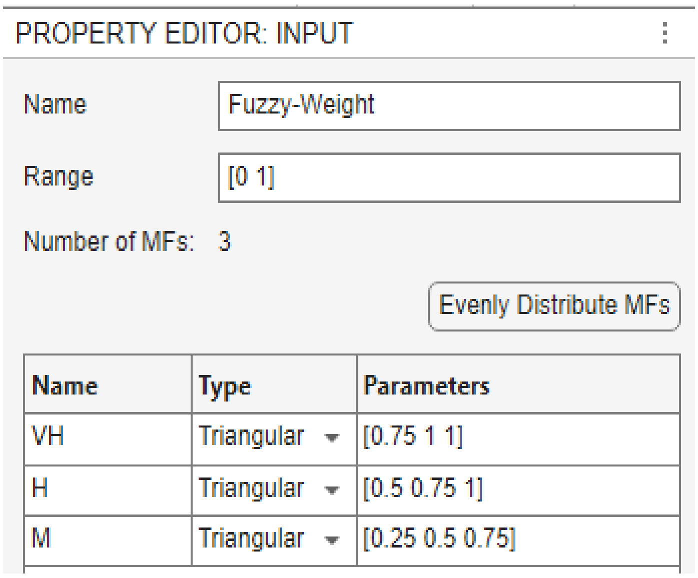


Fig 8.15:Parameter Detail for Input (Fuzzy-Weight): Tester's View

8.3.3 Detail of Input Fuzzy-Rating

The input Fuzzy-Rating includes three triangular membership functions very-high (VH), high (H) and medium (M)illustrated in Fig.8.16 along with the parameters illustrated in Fig.8.17.

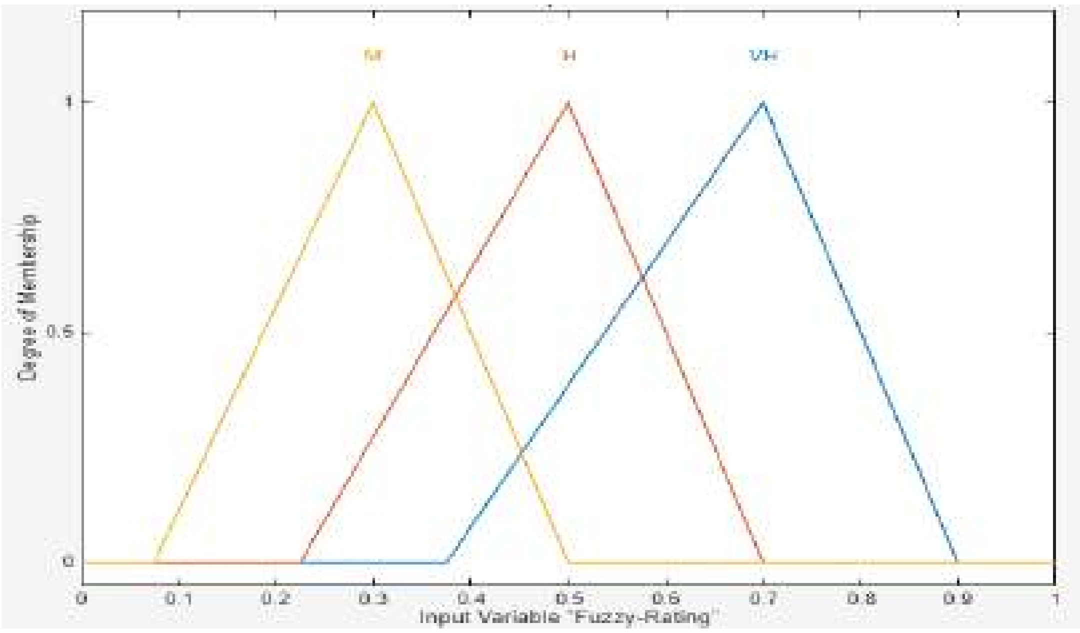
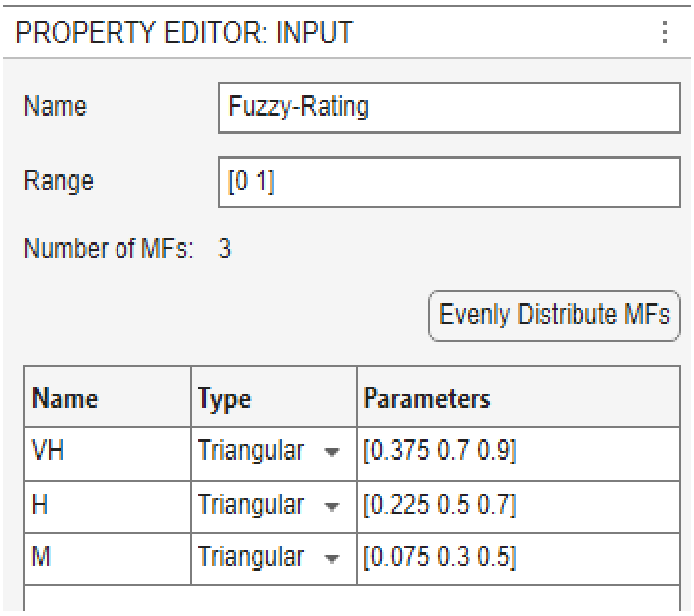


Fig 8.16:Membership Function Plot for Input (Fuzzy-Rating): Tester's View



PROPERTY EDITOR: INPUT

Name: Fuzzy-Rating

Range: [0 1]

Number of MFs: 3

Evenly Distribute MFs

Name	Type	Parameters
VH	Triangular	[0.375 0.7 0.9]
H	Triangular	[0.225 0.5 0.7]
M	Triangular	[0.075 0.3 0.5]

Fig 8.17:Parameter Detail for Input (Fuzzy-Rating): Tester's View

8.3.4 Detail of Output Over-All-Rating-Tester

The outputOver-All-Rating-Testerincludes eight triangular membership functions, very high (VH)to very low (VL) illustrated in Fig.8.18 along with the parameters illustrated in Fig.8.19.

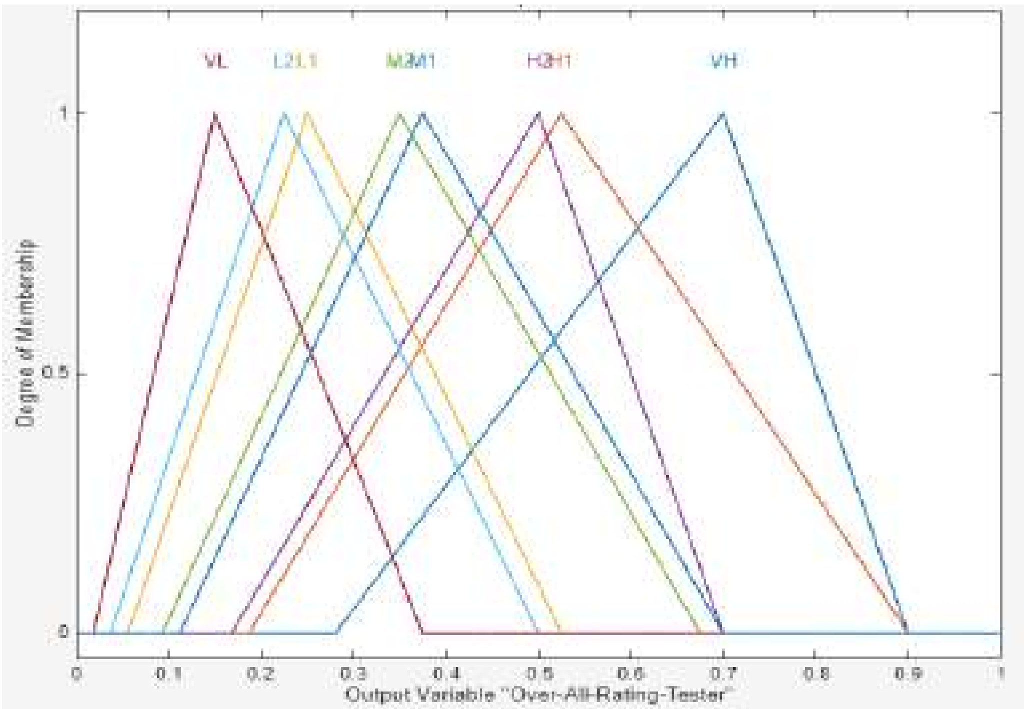


Fig 8.18:Membership Function Plot for Output (Over-All-Fuzzy-Rating): Tester's View

PROPERTY EDITOR: OUTPUT

Name: Over-All-Rating-Tester

Range: [0 1]

Number of MFs: 8

Evenly Distribute MFs

Name	Type	Parameters
VH	Triangular	[0.28125 0.7 0.9]
H1	Triangular	[0.1875 0.525 0.9]
L1	Triangular	[0.05625 0.25 0.525]
H2	Triangular	[0.16875 0.5 0.7]
M2	Triangular	[0.09375 0.35 0.675]
L2	Triangular	[0.0375 0.225 0.5]
VL	Triangular	[0.01875 0.15 0.375]
M1	Triangular	[0.1125 0.375 0.7]

Fig 8.19:Parameter Detail for Output (Over-All-Fuzzy-Rating): Tester's View

8.3.5 Control Surface for Over All Crisp Rating as per Tester

The following figure Fig.8.20, presents a numerical representation of the overall quality of the mobile application prototype currently being developed, based on the perspective of the tester.

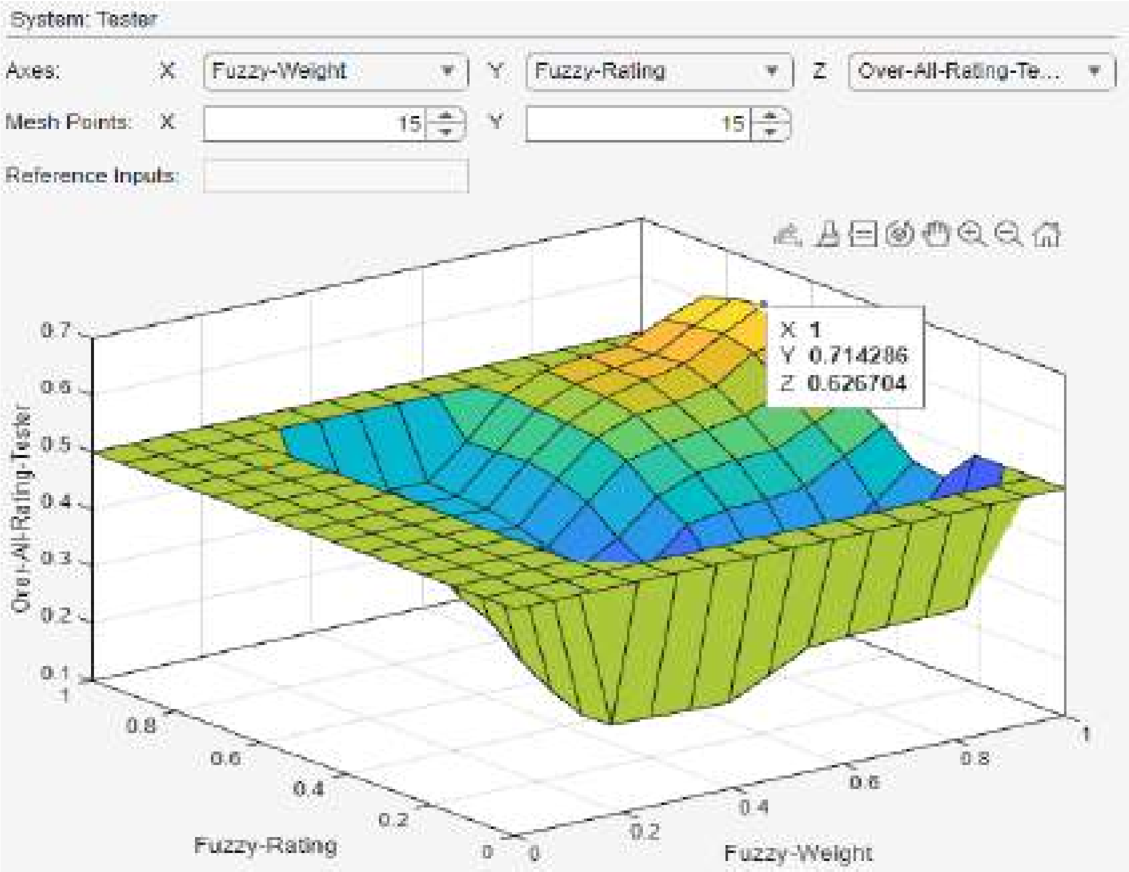


Fig 8.20:Control Surface for Over-All-Rating: Tester's View

The altitude Z is represented by the color of the surface in MATLAB surface plot (control surface), where X, Y and Z are illustrated as shown in Fig 8.20. The color changes as a result of change in the altitude.The tester's view has an overall quality of 0.6280 or 62.80% according to the suggested fuzzy-based mathematical framework MAQM-MA. The MATLAB simulation achieved an overall quality of 0.62670, equivalent to 62.67%. Therefore, the outcome obtained has been confirmed according to the MAQM-MA.

8.4 VALIDATION: β-USER'S VIEW

The following sub-sections will illustrate the procedure of quantitatively validating the overall quality of a mobile application using MATLAB Fuzzy Logic Tool, based on findings from chapter six.

8.4.1 Fuzzy Inference System (FIS)

This section explains about the implementation of FIS for β-User's view with the help of Mamdani Type-1. The FIS includes two fuzzy inputs, fuzzy weight, FW and fuzzy rate, FR and OVER-ALL-RATING-END-USER as output along with fuzzy inference rules as illustrated in Fig.8.21. The property editor explains the property about FIS as illustrated in Fig 8.22 along with sample fuzzy rules as illustrated in Fig 8.23.

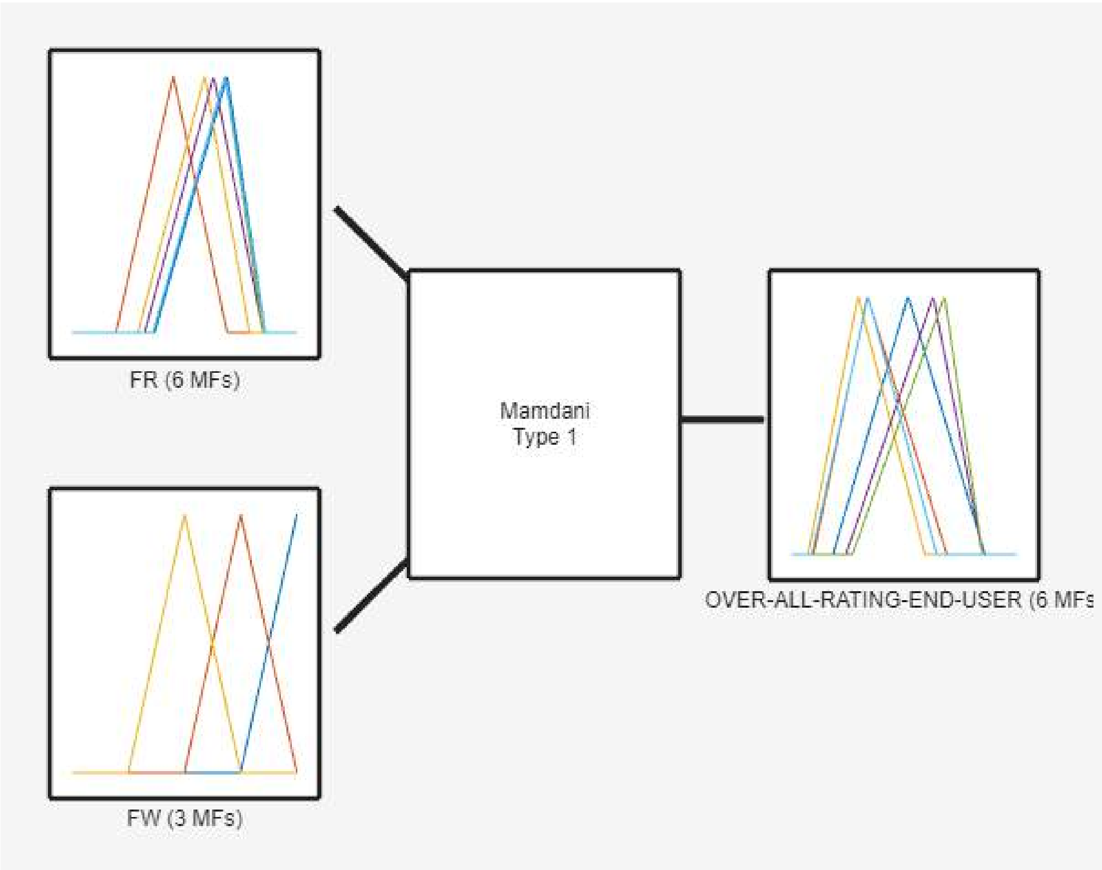


Fig 8.21: Fuzzy Inference System (FIS) for β-User's View

PROPERTY EDITOR: FIS

Type:	Mamdani Type-1
Name	end-user
And method	prod
Or method	max
Implication method	min
Aggregation method	max
Defuzzification method	centroid
Inputs:	2
Outputs:	1
Rules:	6

Fig 8.22: Property Editor for Fuzzy Inference System for β-User's View

	Rule	Weight	Name
1	If Fuzzy-Rating is VH and Fuzzzy-Weight is H then OVER-ALL-RATING-E...	1	rule1
2	If Fuzzy-Rating is VL and Fuzzzy-Weight is H then OVER-ALL-RATING-E...	1	rule2
3	If Fuzzy-Rating is L and Fuzzzy-Weight is M then OVER-ALL-RATING-EN...	1	rule3
4	If Fuzzy-Rating is M and Fuzzzy-Weight is VH then OVER-ALL-RATING-E...	1	rule4
5	If Fuzzy-Rating is H1 and Fuzzzy-Weight is VH then OVER-ALL-RATING-...	1	rule5
6	If Fuzzy-Rating is H2 and Fuzzzy-Weight is M then OVER-ALL-RATING-E...	1	rule6

Fig 8.23: Sample Fuzzy Rules for β-User's View

8.4.2 Detail of Input fuzzy weight (FW)

The input fuzzy weight (FW) includes three triangular membership functions very-high (VH), high (H) and medium (M) illustrated in Fig. 8.24 along with the parameters illustrated in Fig. 8.25.

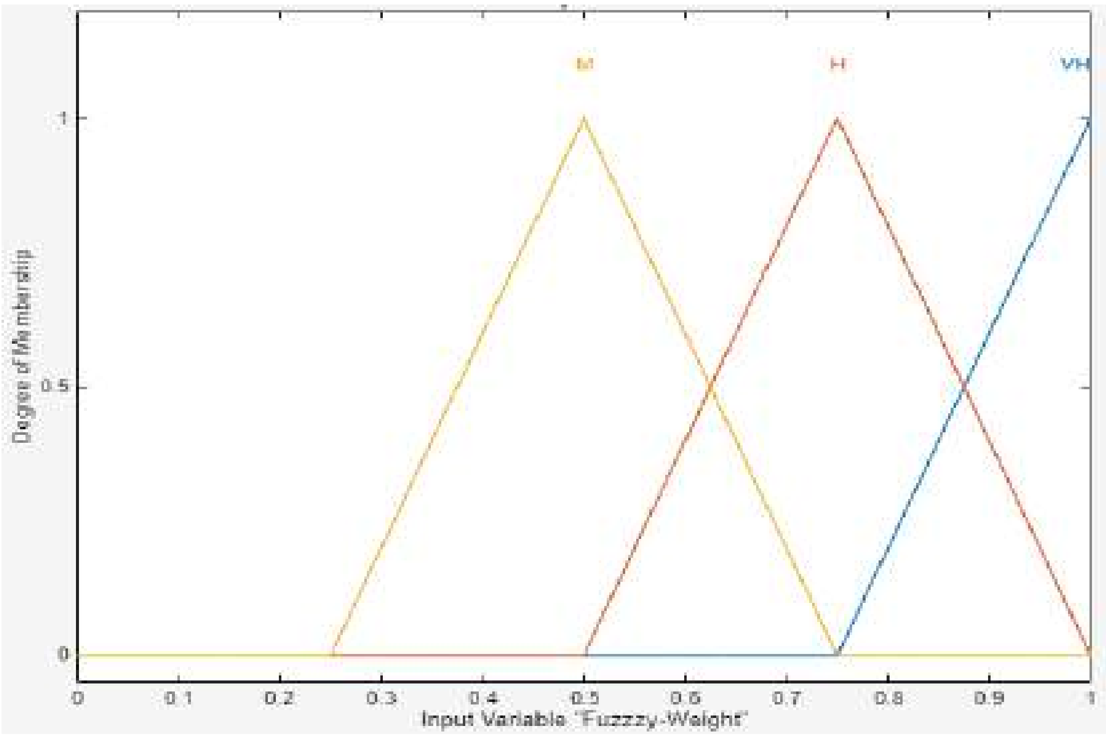


Fig 8.24: Membership Function Plot for Input FW (β-User's View)

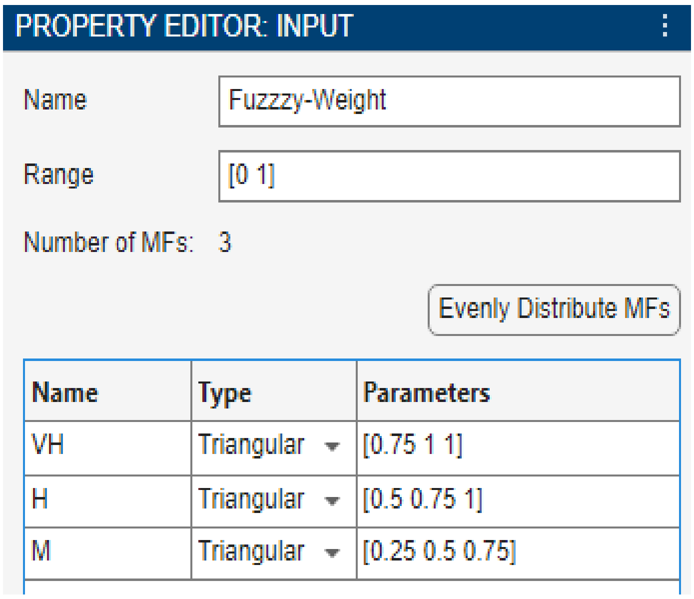


Fig 8.25: Parameter Detail for Input Fuzzy-Weight (β-User's View)

8.4.3 Detail of Input Fuzzy-Rating

The input Fuzzy-Rating includes six triangular membership functions, from very-high (VH) to very-low(VL) illustrated in Fig.8.26 along with the parameters illustrated in Fig.8.27.

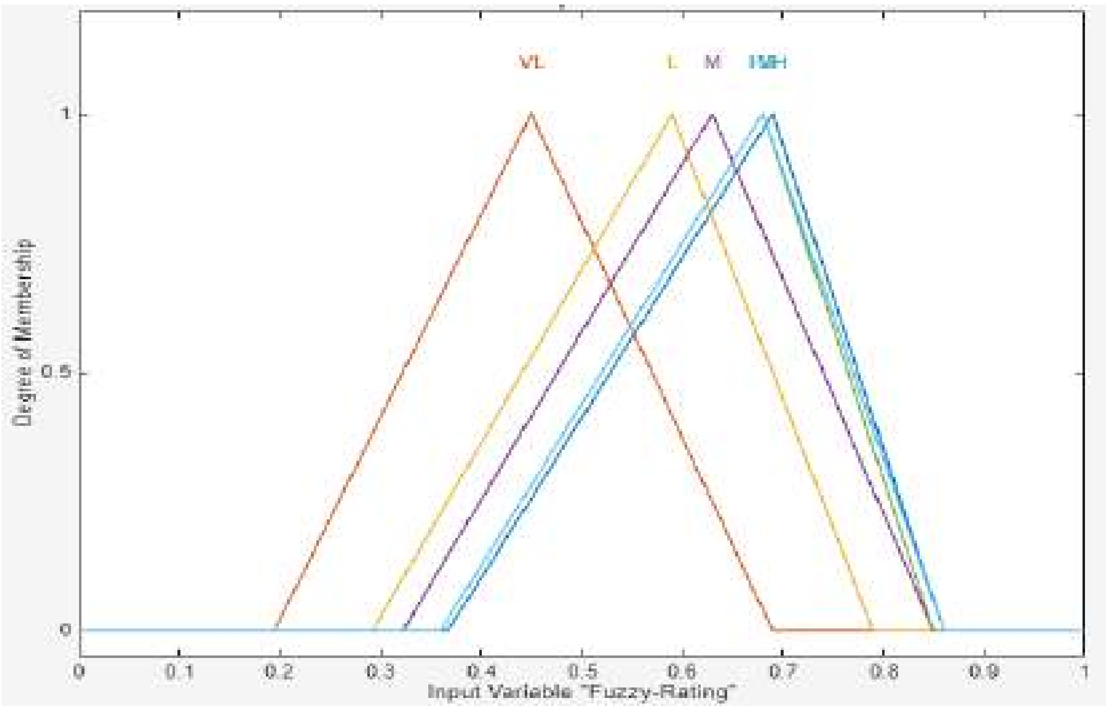
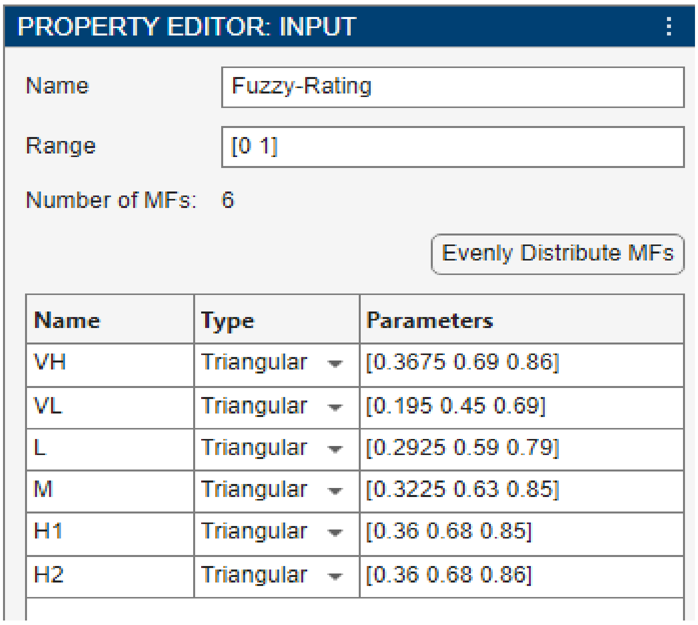


Fig 8.26: Membership Function Plot for Input Fuzzy-Rating (β-User's View)



PROPERTY EDITOR: INPUT

Name: Fuzzy-Rating

Range: [0 1]

Number of MFs: 6

Evenly Distribute MFs

Name	Type	Parameters
VH	Triangular	[0.3675 0.69 0.86]
VL	Triangular	[0.195 0.45 0.69]
L	Triangular	[0.2925 0.59 0.79]
M	Triangular	[0.3225 0.63 0.85]
H1	Triangular	[0.36 0.68 0.85]
H2	Triangular	[0.36 0.68 0.86]

Fig 8.27: Parameter Detail for Input Fuzzy-Rating (β-User's View)

8.4.4 Detail of Output Over-All-Rating-β-User's

The output Over-All-Rating-End-User includes six triangular membership functions, very high (VH) to very low (VL) illustrated in Fig.8.28 along with the parameters illustrated in Fig.8.29.

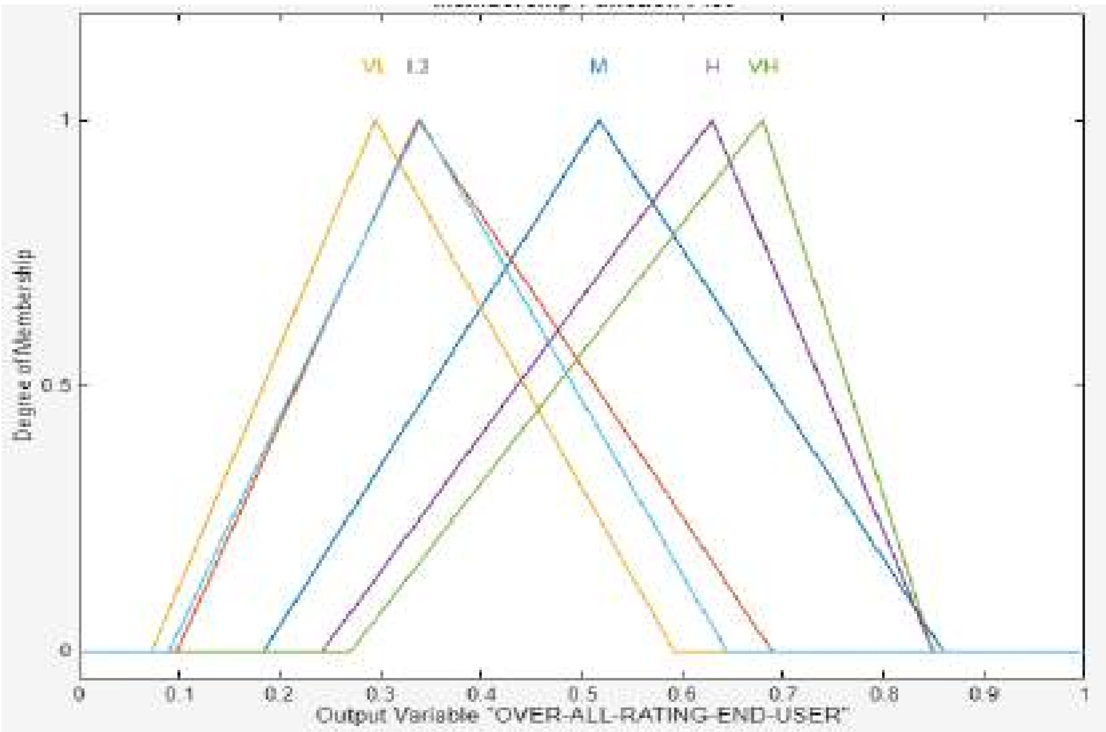


Fig 8.28: Membership Function Plot for Output Over-All-Rating-β-User's View

PROPERTY EDITOR: OUTPUT

Name: OVER-ALL-RATING-END-USER

Range: [0 1]

Number of MFs: 6

Evenly Distribute MFs

Name	Type	Parameters
M	Triangular	[0.1838 0.5175 0.86]
L2	Triangular	[0.0975 0.3375 0.69]
VL	Triangular	[0.0731 0.295 0.5925]
H	Triangular	[0.2419 0.63 0.85]
VH	Triangular	[0.27 0.68 0.85]
L1	Triangular	[0.09 0.34 0.645]

Fig 8.29: Parameter Detail for Over-All-Rating-β-User

8.4.5 Control Surface for Over All Crisp Rating as per β-User

The following Fig.8.30, presents a numerical representation of the overall quality of the mobile application prototype currently being developed, based on the perspective of the β-User .

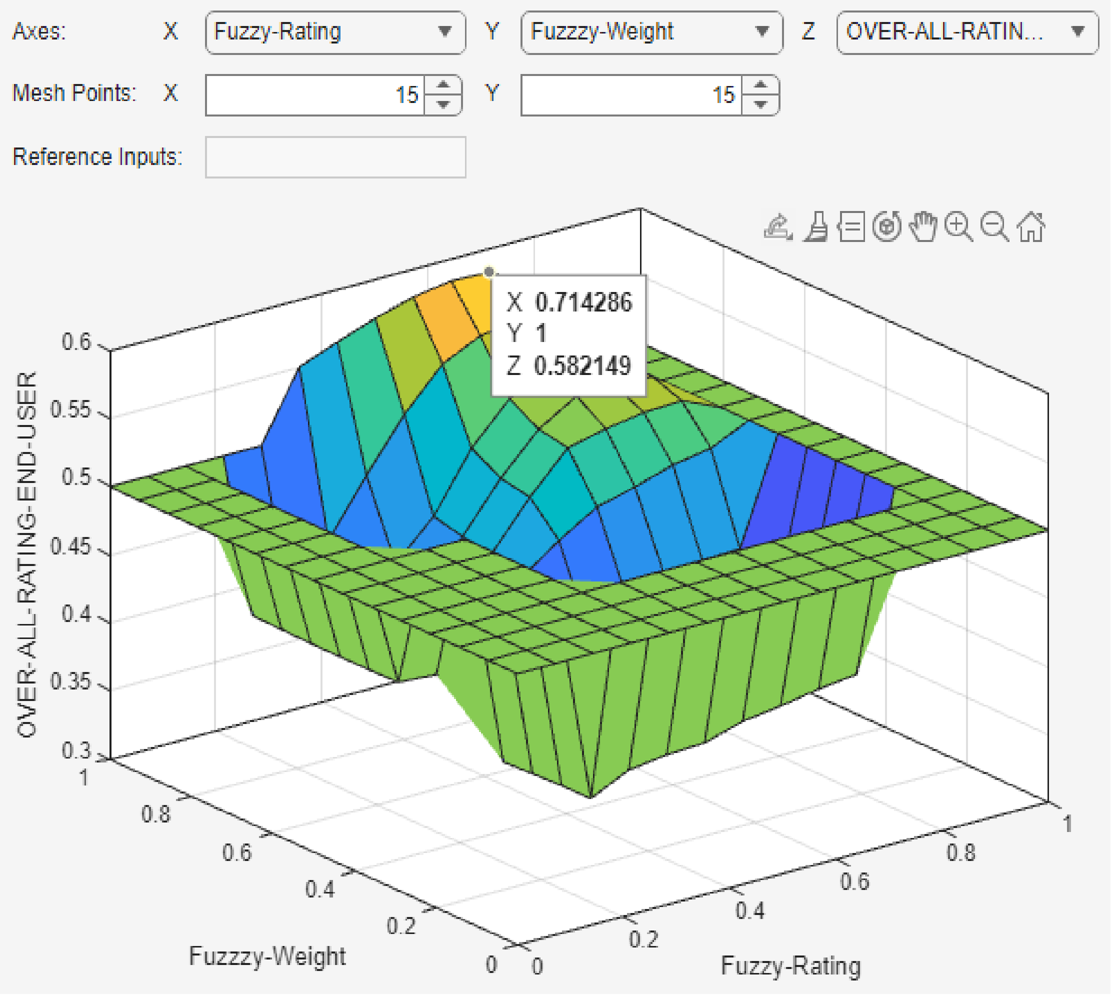


Fig 8.30: Control Surface for Over-All-Rating-β-User with Over-All Crisp Quality 58.21%

The altitude Z is represented by the color of the surface in MATLAB surface plot (control surface), where X, Y and Z are illustrated as shown in Fig 8.30. The color changes as a result of change in the altitude.The overall quality of β-User's view is 0.5875 or 58.75% as per proposed fuzzy based mathematical framework MAQM-MA. The overall quality obtained as per MATLAB simulation is 0.58214 or 58.21 %. Thus the result obtained as per MAQM-MA verified.

8.5 OVER-ALL VIEW

The following sub-sections will illustrate the procedure of quantitatively validating the overall quality of a mobile application using MATLAB Fuzzy Logic Tool, based on findings from chapter six.

8.5.1 Fuzzy Inference System (FIS)

This section explains about the implementation of FIS for three perspectives with the help of Mamdani Type-1. The FIS includes two fuzzy inputs, fuzzy weight, FUZZY-WEIGHT and fuzzy rate, FUZZY-RATE and OVER-ALL-FUZZY-RATING as output along with fuzzy inference rules as illustrated in Fig.8.31. The property editor explains the property about FIS as illustrated in Fig 8.32 along with sample fuzzy rules as illustrated in Fig 8.33.

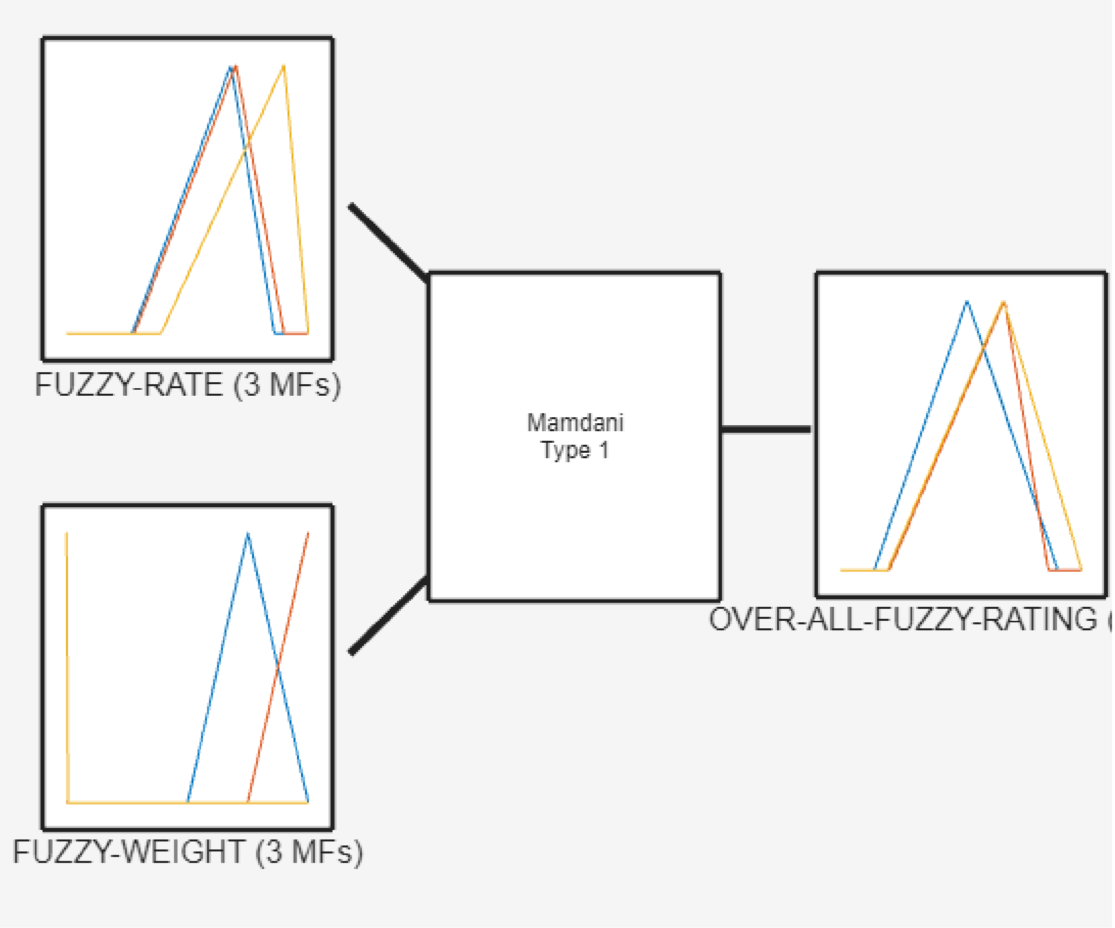


Fig 8.31: Fuzzy Inference System (FIS) for Over-All View

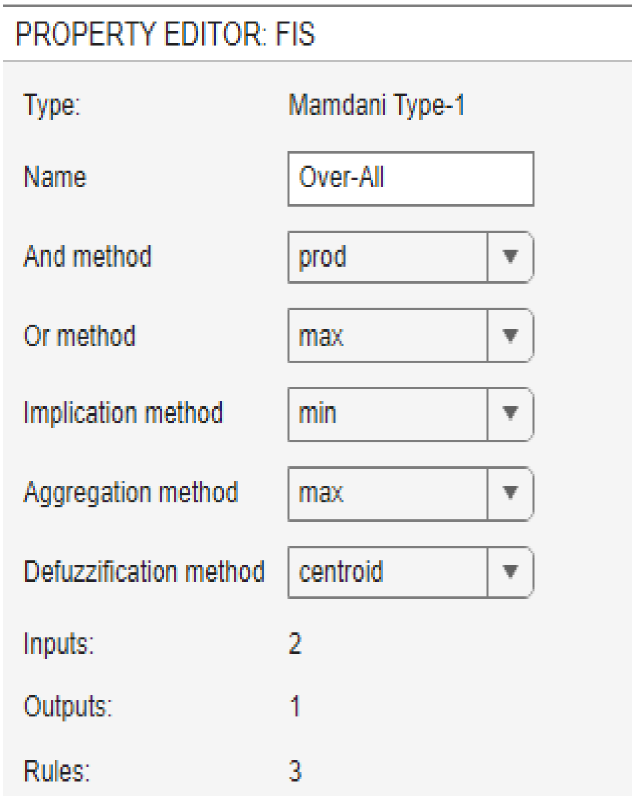


Fig 8.32:Property Editor for Fuzzy Inference System for Over-All View

	Rule	Weight	Name
1	If FUZZY-RATE is Very-High and FUZZY-WEIGHT is High then OVER-ALL...	1	rule1
2	If FUZZY-RATE is High and FUZZY-WEIGHT is High then OVER-ALL-FUZ...	1	rule2
3	If FUZZY-RATE is Medium and FUZZY-WEIGHT is Very-High then OVER-...	1	rule3

Fig 8.33: Sample Fuzzy Rules for Over-All View

8.5.2 Detail of Input Fuzzy-Weight

The input fuzzy weight (FUZZY-WEIGHT) includes two triangular membership functions very-high (VH) and high (H) illustrated in Fig. 8.34 along with the parameters illustrated in Fig. 8.35.

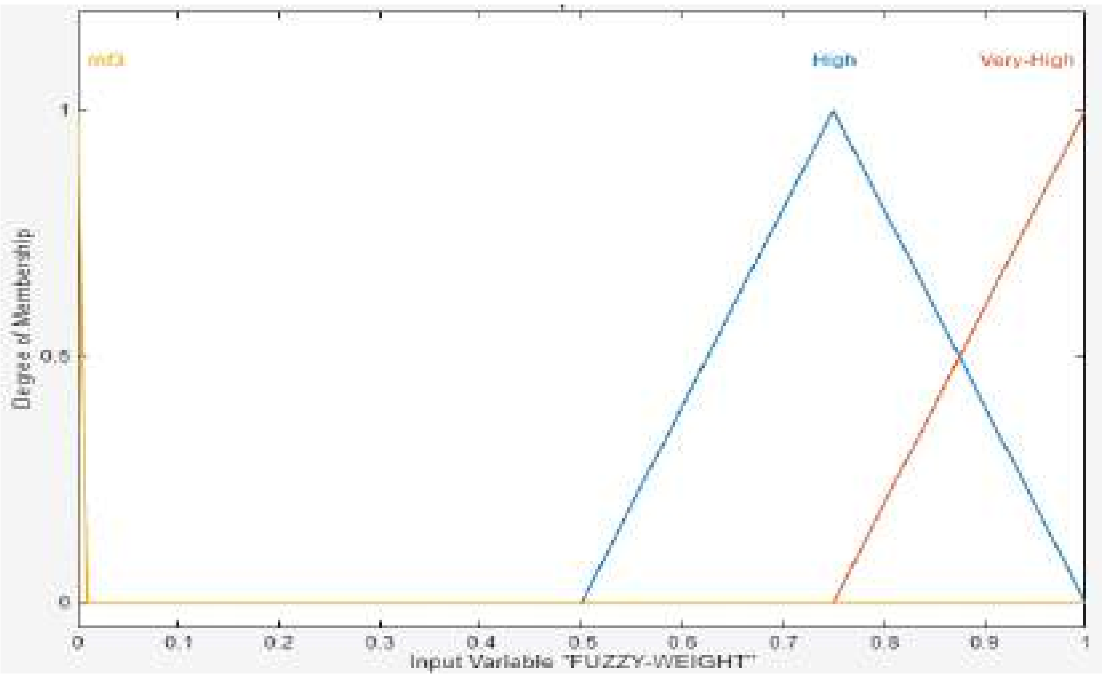
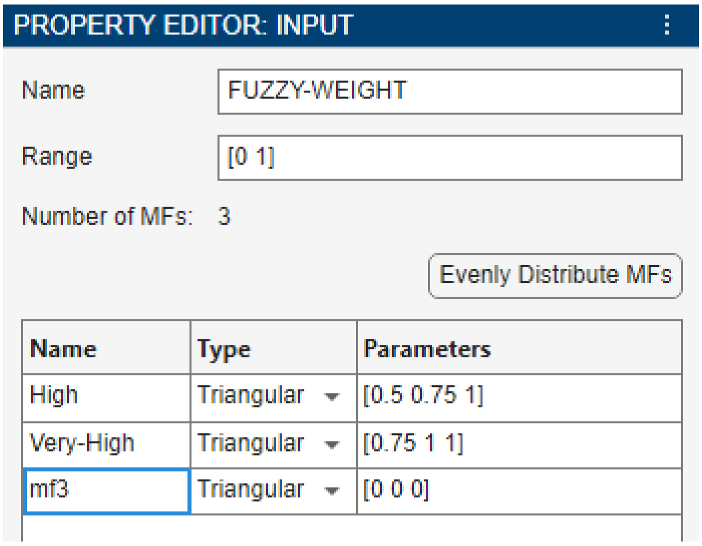


Fig 8.34: Membership Function Plot for Input Fuzzy-Weight



PROPERTY EDITOR: INPUT

Name: FUZZY-WEIGHT

Range: [0 1]

Number of MFs: 3

Evenly Distribute MFs

Name	Type	Parameters
High	Triangular	[0.5 0.75 1]
Very-High	Triangular	[0.75 1 1]
mf3	Triangular	[0 0 0]

Fig 8.35: Parameter Detail for Input Fuzzy-Weight

8.5.3 Detail of Input Fuzzy-Rating

The input Fuzzy-Rating includes three triangular membership functions, very-high (VH) , high (H) and medium (M) illustrated in Fig.8.36 along with the parameters illustrated in Fig.8.37.

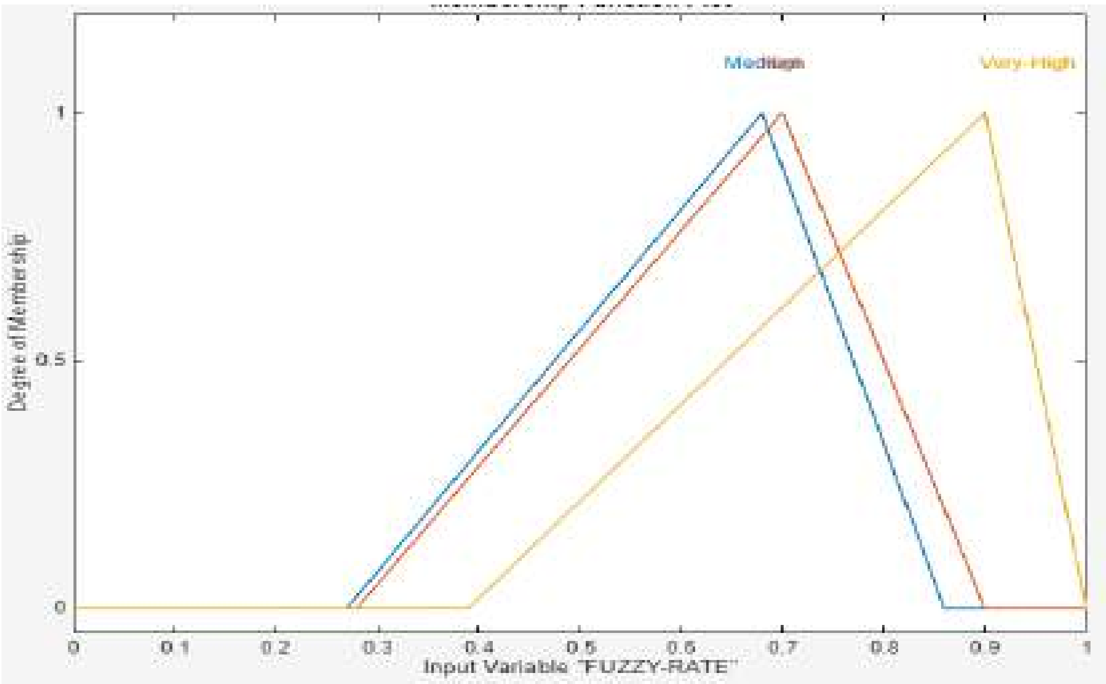


Fig 8.36: Membership Function Plot for Input Fuzzy-Rating

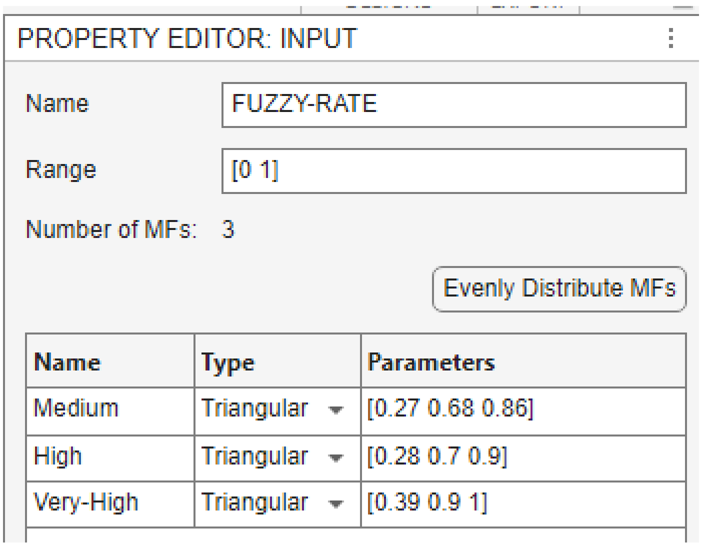


Fig 8.37: Parameter Detail for Input Fuzzy-Rating

8.5.4 Detail of Output

The output OVER-ALL-FUZZY-RATING includes three triangular membership functions, very high (VH), high (H) and medium (M) illustrated in Fig.8.38 along with the parameters illustrated in Fig.8.39.

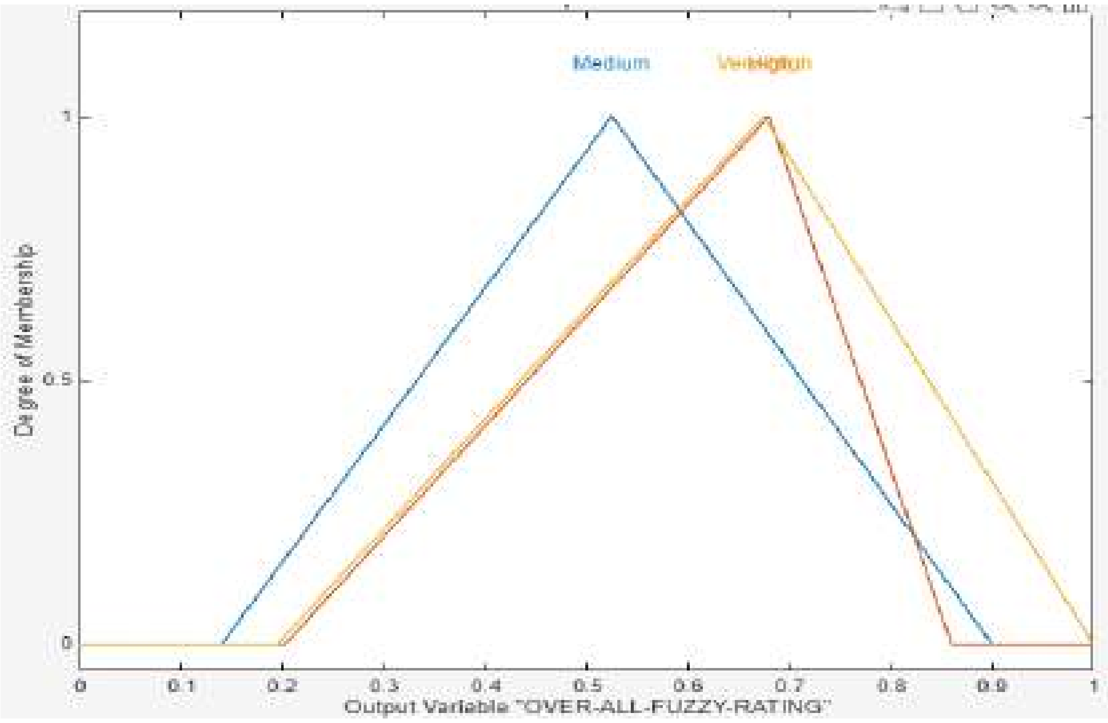
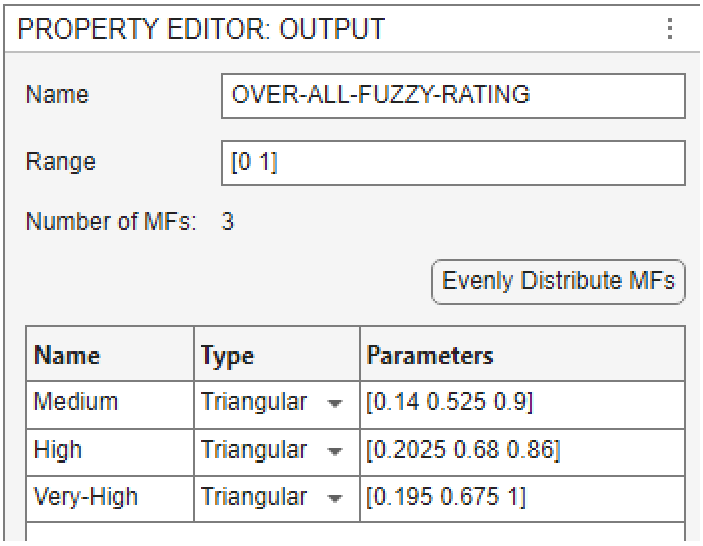


Fig 8.38: Membership Function Plot for Output Over-All-Rating



PROPERTY EDITOR: OUTPUT

Name: OVER-ALL-FUZZY-RATING

Range: [0 1]

Number of MFs: 3

Evenly Distribute MFs

Name	Type	Parameters
Medium	Triangular	[0.14 0.525 0.9]
High	Triangular	[0.2025 0.68 0.86]
Very-High	Triangular	[0.195 0.675 1]

Fig 8.39: Parameter Detail for Over-All-Rating

8.5.5 Control Surface for Over All Crisp Rating as per three perspectives

The following Fig.8.40, presents a numerical representation of the overall quality of the mobile application prototype currently being developed, based on the all three perspective i.e. developer, tester and β-User.

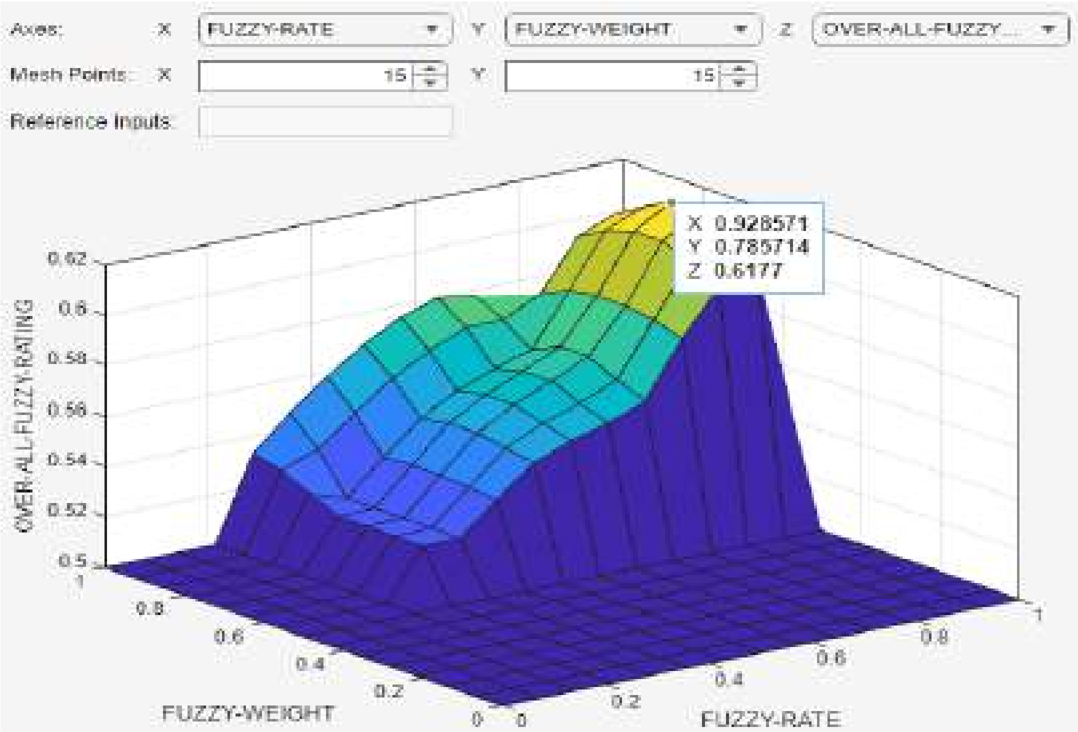


Fig 8.40: Control Surface for Over-All-Rating with Over-All Crisp Quality 61.77%

The altitude Z is represented by the color of the surface in MATLAB surface plot (control surface), where X, Y and Z are illustrated as shown in Fig 8.40. The color changes as a result of change in the altitude.The overall quality of three perspectives is 0.6265 or 62.65% as per proposed fuzzy based mathematical framework MAQM-MA. The overall quality obtained as per MATLAB simulation is 0.6177 or 61.77 %. Thus the result obtained as per MAQM-MA verified.

8.6 SUMMARY

This chapter explains about validation with the help of MATLAB Fuzzy Tool. The process of quantification which involves MAQM-MA framework accepts two fuzzy inputs, fuzzy rate and fuzzy weight which further converge to overall fuzzy rating and overall crisp quality of any domain specific mobile application. As a consequence, the result obtained by the fuzzy-based mathematical framework MAQM-MA discussed in chapter 6 is validated since the simulated outcome produced is identical to the assessment of the framework MAQM-MA.

CHAPTER 9
CONCLUSION& FUTURE SCOPE

9.1 CONCLUSION

The use of mobile applications has grown tremendously in recent years; these applications serve variety of functions.A rising number of users are looking for ways to satisfy their needs in relation to the application itself, such as mobile games, mobile commerce, mobile-based education, and other types of mobile applications. It is very important to ensure that these mobile applications should be created in such a manner which guarantees the high quality of the mobile application. This is done in order to maintain the satisfaction of users and encourage them to use it. Nevertheless, it is challenging to evaluate the subjective and usually vague expectations of β-users once the program has been released. Developers, testers, β-users, and decision makers of the mobile application development industry are constantly investigating apps, for finding out whether the mobile application has been achieved the decided benchmark of quality level or not. These decision makers may include stakeholders, investors, and quality managers in their investigation process. The assessment of qualitative features that are dependent on software quality parameters will make the implementation of this endeavor easier to accomplish. Therefore, it is of the utmost importance to choose the appropriate assessment approach.

The novel contribution of this research study is a comprehensive generic framework MAQM-MA for quantifying software quality factors of mobile applications, which is customizable to specific domain requirements. Theproposed framework has considered perspective of developer, tester and β-user along with implicit view of decision makers in the form of fuzzy weight. The decision makers who are responsible for ensuring that mobile applications are of a suitable quality and thusdecide fuzzy weight. The framework contains suitable quality factors along with relevant metrics for developer, tester and β-user's perspective. Most of the quality factors are common in case of developer and tester but they have their specific purpose which can be differentiated by their metrics.Metrics are designed to serve different purposes along with common quality factor. As per agile methodology of SDLC the usability attribute depends upon perspective of β-user. The customization for the framework of mobile application mainly depends upon metrics designed for β-user's perspective. The β-user perspective is very important that converges in to popularity and branding for any mobile application which has to be observed by the decision makers of mobile application development business.

The quality factors are based upon two qualitystandards ISO/IEC-9126 and ISO/IEC-9241-11 in thisresearch study.There are two inputs that are used in the evaluation, fuzzy rate and fuzzy weight. The assessment of a fuzzy rating system with five scale fuzzy rating is based on surveys or questionnaires as per developed prototype of mobile application. The term "fuzzy weight" refers to the relative relevance of various factors of software quality. This research study presents a novel application of pair-wise comparison in the form of fuzzy based mathematical algorithm FW-MA discussed in section 5.3.2. The purpose of this application is to determine the proper fuzzy weight for software quality factor of mobile applications while they are being developed as per the mutual concern of decision makers. This will be of great assistance to the mobile application business in terms of generating and analyzing fuzzy weights in order to strengthen the process that converts qualitative to quantitative features during the development of any mobile application. The proposed MAQM-MA framework is also implemented using a tool developed in java script and Node.js run-time.

Characteristics of proposed fuzzy based mathematical framework MAQM-MA:

1. This framework includes two quality standards ISO/IEC-9126, for developer, tester and ISO/IEC-9241-11, for β-Users.
2. This framework is customizable for any mobile application for developer and tester.
3. The usability assessment depends upon β-Users as per the domain of mobile application.

4. This framework includes a novel fuzzy based mathematical algorithm FW-MA, which evaluates appropriate fuzzy weight for software quality factors according to decision makers.
5. This framework accepts two inputs fuzzy weight and fuzzy rate. Quantifies crisp quality of mobile application during its development. The quantification may be used for feedback and rectification.

This is a novel framework which includes all entities of a mobile application development organization and monitors the quality of a mobile app during its development.

The overall results obtained by proposed framework MAQM-MA along with fuzzy based mathematical algorithm FW-MA and validated by MATLAB Fuzzy Tool are illustrated in table 9.1 and Fig 9.1.

Table 9.1: Comparison of Result: MAQM-MA (Calculated Result) and MATLAB Fuzzy Tool (Simulated Result)

Views	**Calculated-Result (MAQM-MA)**	**Simulated-Result (MATLAB Fuzzy Tool)**
Developer's View	75.89 %	75.59 %
Tester's View	62.8 %	62.67 %
β-User's View	58.75 %	58.21 %
Over-all View	62.65 %	61.77 %

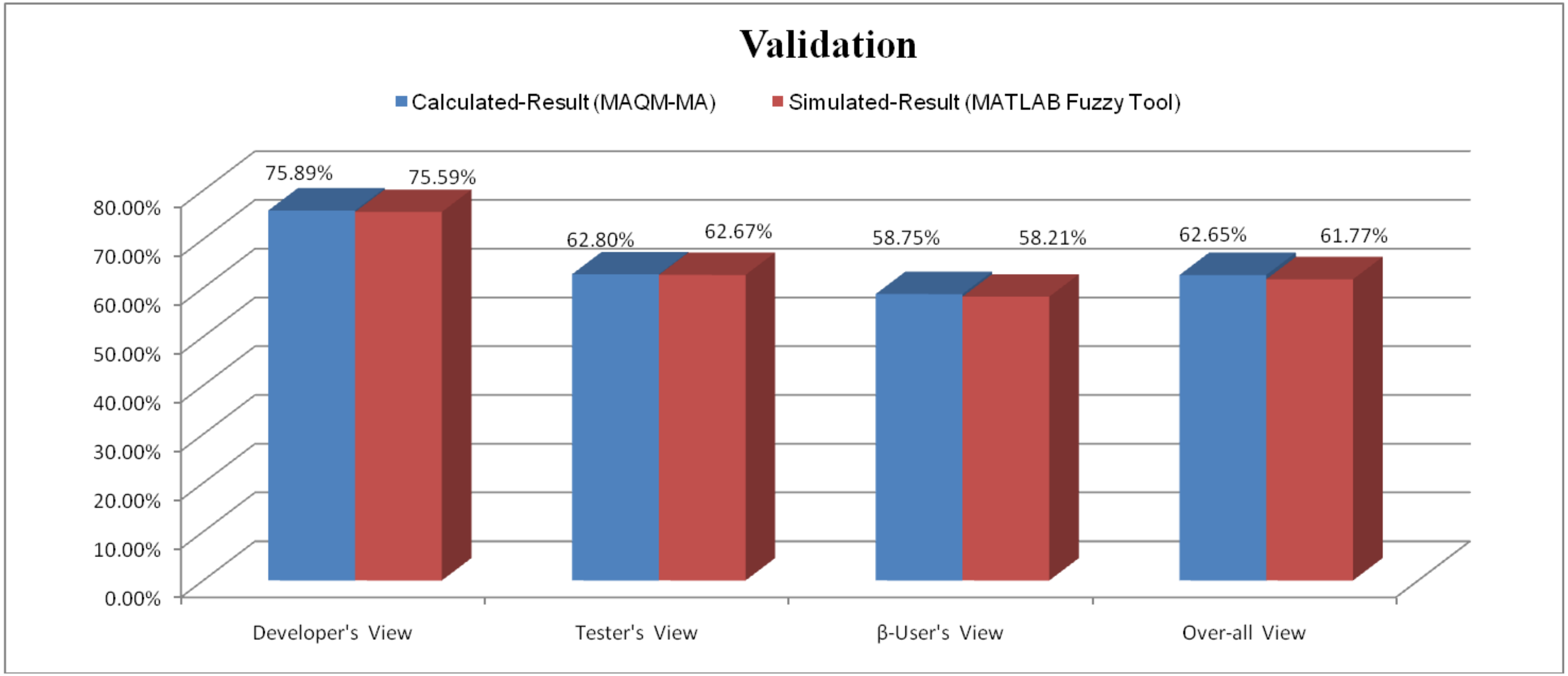


Fig. 9.1: Validation as per MATLAB Fuzzy Tool

9.2 RECOMMENDATIONS FOR MOBILE APPLICATION BUSINESS

One of the most crucial aspects of every research is the analysis of the data collected during research work. This analysis can be carried out with the use of numerical data, quality is a subjective field, and the only data that can be used for deductive reasoning is numerical data. For this aim, MAQM-MA is used. Mobile application industry have the option of using this strategy in order to quantify domain-specific mobile applications, assess and forecast the numerical data, In order to guarantee that judgments are taken on an objective basis. As a result, it is useful in assessing which areas need improvement, rating the significance of additions or updates, and enhancing the effectiveness of the development process. Establishing precise goals and keeping track of progress over the course of time is made feasible as a result of this.

Mobile application industry may make use of methods like surveys, user analytics, user testing, and feedback as a tool for the purpose of converting qualitative framework into quantitative data.The evaluation of fuzzy ratings is the responsibility of each of these methodological techniques. The relevance of fuzzy weight is dependent on the decision makers reaching an agreement for it to be considered significant. Consequently, the accuracy of the calculation depends upon the value of the fuzzy weight's value. Accuracy may be improved with the use of fuzzy AHP. The attainment of accurate evaluation will be facilitated by this, which will provide an acceptable basis. Because of this, it is strongly suggested to all of the businesses that are involved in the production of mobile apps that this framework be used in order to achieve acceptable computation.

This will ensure that companies who produce mobile applications are able to provide a full understanding of the performance of their app as well as the satisfaction of its users via the use of trustworthy quantitative data. Therefore, it will increase the level of confidence of investors and stakeholders in the mobile application development business.

9.3 FUTURE SCOPE

The present research might be brought into the modern day by using artificial intelligence or machine learning. It is possible to train and apply the data produced from this research for cluster analysis, which may be used for a variety of decision-making strategies. A fuzzy cognitive map is a tool that may be used to assess the influence of overall quality parameters on a system. As a result, changes in one quality element will have an influence on another quality factor, and this will allow us to forecast the behavior of the overall quality perception as a developer, tester, and β-user. Automation has the potential to function as a novel approach, as well as to optimize time and financial restrictions. In order to get the best possible outcome, it is possible to include both manual and automated processes.

SUMMARY

The proposed research work takes its motivation to develop a fuzzy based mathematical framework to access the overall quality of mobile apps might originate from a number of various sources. The proposed framework will assign numerical ratings for the quality of mobile apps. So far, there is no mathematical framework available that could help developers and testersto numerically access the quality of their apps and make improvements. This leads to the fact that there is a significant need for such a framework. Evaluation in numeric form can also be used to facilitate comparisons between different applications and the identification of areas for improvement.

This research is conducted using qualitative and quantitative research methodologies. Fuzzy questionnaire based on five scales for developers, testers and β-users to calculate the fuzzy rating for the metrics as designed. A survey was conducted on prototype of M-Commerce application and the questionnaire was filled by developer, tester and β-users to determine the fuzzy rating for each metric. These fuzzy ratings along with the fuzzy weight provided by decision maker, serve as input to a proposed fuzzy based mathematical framework.

The novel contribution of this research study is a comprehensive generic framework MAQM-MA for quantifying software quality factors of mobile applications, which are customizable to specific domain requirements. The software quality factors are based upon two quality standards ISO/IEC-9126 and ISO/IEC-9241-11 in this research study. There are two inputs that are used in the evaluation: fuzzy rate and fuzzy weight. The proposed framework has considered perspective of developer, tester and β-user along with implicit view of decision makers in the form of fuzzy weight. The decision makers who are responsible for ensuring that mobile applications are of a suitable quality and thus decide fuzzy weight. The framework contains suitable quality factors along with relevant metrics for developer, tester and β-user's perspective. This framework calculates the overall fuzzy and crisp quality according to developer, tester, β-users and their combined impact. Thus, the overall

quality of the mobile application from these three perspectives is quantitatively evaluated using the proposed framework.

This research study presents a novel technique of pair-wise comparison in the form of fuzzy based mathematical algorithm FW-MA. The purpose of this is to determine the proper fuzzy weight for software quality factor of mobile applications while they are being developed as per the mutual concern of decision makers. This will be of great assistance to the mobile application business in terms of generating and analyzing fuzzy weights in order to strengthen the process that converts qualitative to quantitative features during the development of any mobile application. The proposed MAQM-MA framework is also implemented using a tool developed in java script and Node.js run-time.

Mobile application industry may use the proposed technique to quantify domain-specific mobile applications, assess and forecast the numerical data, in order to guarantee that judgments are taken on an objective basis. As a result, it is useful in assessing which areas need improvement, rating the significance of additions or updates, and enhancing the effectiveness of the development process.

This will ensure that companies who produce mobile applications are able to provide a full understanding of the performance of their app as well as the satisfaction of its users via the use of trustworthy quantitative data. Therefore, it will increase the level of confidence of investors and stakeholders in the mobile application development business.

BIBLIOGRAPHY

[1] ISO/IEC 9000:2015, "Quality management- Fundamentals and vocabulary", international organization for standardization, 2015.

[2] Rafi Ahmad Lone, Mushtaq Ahmad Bhat, "Impact of Product Quality on Customer Satisfaction", Evidence from Selected Consumer Durables, 2023.

[3] Y. A. Storey, R. Briggs, H. Jones, R. Russell, Chapter 4: QUALITY ASSURANCE, 2011.

[4] Dr.Jagannath Singh, NigussuBitewKassie, "User's Perspective of Software Quality", ICECA, 2018.

[5] GaganGurung, Rahul Shah, Dhiraj Prasad Jaiswal, "Software Development Life Cycle Models-A Comparative Study", International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2020.

[6] Dawn Owens, Deepak Khazanchi, "Software Quality Assurance", 2009.

[7] DimitrisStavrinoudis, MichalisXenos, "Comparing internal and external software quality measurements", Frontiers in Artificial Intelligence and Applications, 2008.

[8] Imran Khan, Linda Hollebeek, MobinFatma, Jamid Islam, Raouf Ahmad, ShadmaShahid, Valdimar Sigurdsson, "Mobile app vs. desktop browser platforms: The relationships among customer engagement, experience, relationship quality and loyalty intention", Journal of Marketing Management. pp.1-23, 2022.

[9] Victor R. Basili, "Software development:A paradigm for the future", In Proceedings ofthe 13th Annual International Computer Software &ApplicationsConference, pp.471-485, Sept 1989.

[10]Victor R. Basili, "Quantitativeevaluationofsoftwaremethodology",https://www.cs.umd.edu/~basili/publications/proceedings/P29.pdf, 1985.

[11] Jim A. McCall, Paul K. Richards, Gene F. Walters, "Factors in Software Quality", Vol.I, II, and III, US Rome Air Development Center Reports, U. S. Department of Commerce, 1977.

[12] R. G. Dromey, "A model for software product quality", IEEE Transactions on Software Engineering, Vol.21, No.2, pp.146-162, February 1995.

[13] Sandra A. Slaughter, Donald E. Harter, Mayuram S. Krishnan, "Evaluating the Cost of Software Quality", Communications of the ACM, Vol.41, No.8, pp.67-73, August, 1998.

[14] Luigi Buglione, Alain Abran, "A Quality Factor for Software",Proceedings of QUALITA99, 3rd International Conference on Quality and Reliability, Paris, France, 1999.

[15] ISO/IEC 9126-1:2001, "Software Engineering-Product Quality—Part 1: Quality Model", International Organization for Standardization, 2001.

[16] OlfaLamouchi, Amar R.Cherif, Nicole LÈvy, ", measurements for evaluating an IS quality", Proceedings of the fifth on Asia-Pacific conference on conceptual modelling, Wollongong, NSW, Australia, pp.39-47, January, 2008.

[17] Yue Jiang, BojanCukic, Tim Menzies, Nick Bartlow, "Comparing design and code metrics for software quality prediction", Proceedings - International Conference on Software Engineering, 2008.

[18] YiannisKanellopoulos, PanosAntonellis, Dimitris Antoniou, Christos Makris, EvangelosTheodoridis, Christos Tjortjis, Nikos Tsirakis, "Code Quality Evaluation

Methodology Using The ISO/IEC 9126 Standard," International Journal of Software Engineering & Applications (IJSEA), Vol.1, No.3, pp.17-36, July 2010.
[19] ISO/IEC 25022:2016, "Systems and software engineering- systems and software quality requirements and evaluation (SQuaRE)-Measurement of quality in use", international organization for standardization, 2016.

[20] Ming-Chang Lee, "Software Quality Factors and Software Quality Metrics to Enhance Software Quality Assurance", British Journal of Applied Science & Technology, 2014.

[21] Rana Muhammad Ammad, MuzamailRafique, SaleemZubair Ahmed, "Study on Software Quality Factors and Metrics to Enhance Software Quality Assurance", International Journal of Study on Software Quality Factors and Metrics to Enhance Software Quality Assurance, Volume 1, Issue 1, March-2021.

[22] B. W. Boehm, J. R. Brown, M. L. Lipow, "Quantitative Evaluation of Software Quality", Proceedings of the 2nd International Conference on Software Engineering, pp.592-605, October 1976.

[23] Victor R. Basili, "Software modeling and measurement:The Goal/Question/Metric paradigm",1992.

[24] Robert B. Grady, Deborah L. Caswell, "Software metrics:Establishingacompany-wideprogram",Prentice-Hall, 1987.

[25] MaryolyOrtega , Maria A. Perez, Teresita Rojas, "Construction of a Systemic Quality Model for Evaluating Software Product", Software Quality Journal, Vol.11, No.3, pp.219-242, 2003.

[26] J. R. Brown, M. Lipow, "Testing for Software Reliability", Proceedings of the international conference on Reliable software, pp.518-527, June 1975,.

[27] Ronan Fitzpatrick, Catherine Higgins, “Usable Software and its Attributes: A synthesis of Software Quality European Community Law and Human-Computer Interaction”, Proceedings of the HCI'98 Conference, Springer, pp.1-19, 1998.

[28] ISO/IEC 9241-11, “Ergonomics of human-system interaction-Part 11: Usability: Definitions and concepts”, international organization for standardization, 2018.

[29] J. Nielsen, "Iterative user-interface design" , in Computer, vol. 26, no. 11, pp. 32-41, Nov. 1993.

[30] Nigel Bevan, “Usability is Quality of Use”, Proceedings of the 6th International Conference on Human Computer Interaction, Elsevier, July 1995.

[31] Manuel F. Bertoa, Antonio Vallecillo, “Usability metrics for software components”, Proceedings of Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE), pp.136-143, April 2006.

[32] T.J. Ross, “Fuzzy Logic with Engineering Applications”,2nd Ed, Wiley India Pvt. Ltd, 2004.

[33] ReenaDadhich, BhaveshMathur, “Measuring Reliability of an Aspect Oriented Software Using Fuzzy Logic Approach”, International Journal of Engineering and Advanced Technology (IJEAT), Volume-1 Issue-5, June 2012.

[34] K. K. Aggarwal, Y. Singh, P. Chandra, M. Puri, “Measurement of Software Maintainability Using a Fuzzy Mode”, Journal of Computer Sciences, Vol.1, No.4, pp.538-542, 2005.

[35] James Senior, Ian Allison, Jonathan A. Tepper, “Automated Software Quality Visualisation Using Fuzzy Logic Techniques”, Communication of the IIMA, Vol.7, No.1, pp.25-40, 2007.

[36] Harish K. Mittal, Pradeep K. Bhatia, PuneetGoswami, “Software Quality Assessment Based on Fuzzy Logic Technique”, International Journal of Software Computing Applications, Issue 3, pp.105-112, 2008.

[37] Deepak Gupta, Vinay Kr. Goyal, Harish Mittal, “Comparative Study of Soft Computing Techniques for Software Quality Model”, International Journal of Software Engineering Research & Practices, Vol.1, No.1, pp.33-37, Jan 2011.

[38] SaumendraPattnaik, Binod Kumar Pattanayak, SrikantaPatnaik, “Software Quality Prediction Using Fuzzy Logic Technique”, International Journal of Information Systems in the Service Sector, pp.51-71, 2019.

[39] T. L. Saaty, “The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation”, Mcgraw-Hill, 1980.

[40] J.J. Buckley, “Fuzzy hierarchical analysis. Fuzzy Sets and Systems”, pp. 233–247, 1985.

[41] ChristerCarlsson, Robert Fuller, “Fuzzy multiple criteria decision making: Recent developments”, Fuzzy Sets and Systems, Vol.78, pp.139-153, 1996.

[42] Salvatore Greco, Benedetto Matarazzo, Roman Slowinski, “The use of rough sets and fuzzy sets in MCDM”, Multicriteria Decision Making.,pp.397-455, 1999.

[43] Michel Grabisch, Marc Roubens, “Application of the Choquet Integral in Multicriteria Decision Making”, Fuzzy measures and integrals, PhysicaVerlag, Berlin, pp.348-374, 2000.

[44] GulcinBuyukozkan, CengizKahraman, Da Ruan, “A fuzzy multi-criteria decision approach for software development strategy selection”, International Journal of General Systems, Vol.33, No.(2-3), pp.259-280, 2004.

[45] Ajit P. Singh and A. K. Vidyarthi, “Optimal allocation of landfill disposal site: A fuzzy multi criteria approach”, Iranian Journal of Environmental Health Science & Engineering, Vol.5, No.1, pp.25-34, 2008.

[46] Praveen RanjanSrivastava, Puneet Jain, A. P. Singh, G. Raghurama, “Software quality factor evaluation using Fuzzy multi-criteria approach”, Proceedings of the 4th Indian International Conference on Artificial Intelligence (IICAI 2009), pp.1012-1029, December 2009.

[47] Sanjay K. Dubey, Disha Sharma, “Software Quality Appraisal Using Multi-Criteria Decision Approach”, International Journal of Information Engineering and Electronic Business, ijieeb, 2015.

[48] FarhadLotfi, TofighAllahviranloo, WitoldPedrycz, MohammadrezaShahriari, “Weight Determination Methods in Fuzzy Environment In: Fuzzy Decision Analysis: Multi Attribute Decision Making Approach”, Studies in Computational Intelligence, Springer, 2023.

[49]Srivastava P.R., K. Kumar K., “An Approach towards Software Quality Assessment”, Communications in Computer and Information Systems Series (CCIS Springer Verlag), Vol.31, No.6, pp.345-346, 2009.

[50] Praveen RanjanSrivastava, AjitPratap Singh, Vageesh K. V., “Assessment of Software Quality: A Fuzzy Multi – Criteria Approach”, Evolution of Computation and Optimization Algorithms in Software Engineering:Applications and Techniques, IGI Global USA, chapter - 11, pp.200-219, 2010.

[51] AzhamHussain, Maria Kutar “Usability Metric Framework for Mobile Phone Application”, 2009.

[52]EulerHortaMarinho, Rodolfo Ferreira Resende,“Quality Factors in Development Best Practices for Mobile Applications”, pp.632-645, 2012.

[53] Ali Idri, KarimaMoumane, Alain Abran, “On the Use of Software Quality Standard ISO/IEC9126 in Mobile Environments”, Proceedings - Asia-Pacific Software Engineering Conference, 2013.

[54] Maximilian Wich, Tommi Kramer, “Enhanced Human-Computer Interaction for Business Applications on Mobile Devices: A Design-Oriented Development of a Usability Evaluation Questionnaire”, Proceedings of the Annual Hawaii International Conference on System Sciences. 2015.

[55] S. Panichella, A. Di Sorbo , E. Guzman, C. A.Visaggio , G. Canfora, H. C. Gall, “How Can I Improve My App?”, Classifying User Reviews for Software Maintenance and Evolution, 2015.

[56] ZuhairElkheir, Ariffin Abdul Mutalib, “Mobile Learning Applications Designing Concepts and Challenges: Survey”, Research Journal of Applied Sciences Engineering and Technology, pp.438-442,2015.

[57] André Nitze, Andreas Schmietendorf, “A survey on mobile users' software quality perceptions and expectations”, ICSTW, 2015.

[58] PauliusPaskevicius, RobertasDamasevicius, “Design of Usable Interface for a Mobile e-Commerce System”, 2016.

[59] Lawrence Barnett, Harvey Carlo & Christos Gatzidis, “First Time User Experiences in Mobile Games: An Evaluation of Usability”, j.entcom, 2018.

[60] A. Hussain, E. O. C. Mkpojiogu, J. Musa, S. Mortada, W. S. Yue., “Mobile Experience Evaluation of an e-Reader App”, Journal of Telecommunication, Electronic and Computer Engineering, pp.11-15, 2018.

[61] S. Shafiq, T. A. Khan,“Role& value of usability in educational learning via game based apps”, International Journal of Scientific and Technology Research, pp.70-77, 2018.

[62] David Parsons, HokyoungRyu, “A Framework for Assessing the Quality of Mobile Learning”, 2006

[63] KarimaMoumane, Ali Idri, AlianAbran, “Usability evaluation of mobile applications using ISO/IEC 9241 and ISO/IEC 25062 standards”, in Springer Plus, vol. 5, no. 1, Springer, pp.1–15, 2016.

[64] Fernando Rosell-Aguilar, “State of the app: A taxonomy and framework for evaluating language learning mobile applications”, *CALICO J.*, vol. 34, no. 2, pp. 243–258, 2017.

[65] SerenBasaran, Firass El Homsi, “Mobile Mathematics Learning Application Selection using Fuzzy TOPSIS”, Int. J. Adv. Comput. Sci. Appl., vol. 13, no. 2, 2022.

[66] Paula Bitrián, Isabel. Buil, Sara Catalán, “Enhancing user engagement: The role of gamification in mobile apps,” J. Bus. Res., vol. 132, pp. 170–185, 2021.

[67] Manish Mishra, ReenaDadhich, “How to Quantify Software Quality Factors for Mobile Applications?: Proposed Criteria”, Emerging Trends in Expert Applications and Security, ICE-TEAS-2023, Lecture Notes in Networks and Systems, vol 681, Springer, Singapore, 2023.

[68] Manish Mishra, ReenaDadhich, “Fuzzy Logic-Based Quantification of Usability Expectation for M-Commerce Mobile Application by Using GQM and ISO/IEC 9241-11”, Journal of Computer Science, 20(1), pp.1-9, 2024.

[69] Manish Mishra, ReenaDadhich,“A Fuzzy AHP Approach to Evaluating the Fuzzy Weight of Mobile Game Application Software Quality Factors: Usability Expectations for Novice Users”, Archives of Advanced Engineering Science (AAES), pp.1–9, 2024.

[70] HaiqiFeng, Tamara Hogler,Wolffried Stucky, “Exploringthecriticalsuccessfactorsformobilecommerce”,In2006InternationalConferenceonMobile Business, IEEE, 2006.

[71] MengtianCui ,Libo Zhu, “Usability Evaluation Methods of User Interface Based on Mobile Games Using Fuzzy Methods”, pp.124-131, 2017.

[72] PawelWeichbroth , “Usability of Mobile Applications: A Systematic Literature Study”, in IEEE Access, vol. 8, 2020.

[73] SorinNădăban, SimonaDzitac, IoanDzitac, “Fuzzy TOPSIS: A General View”, Procedia Computer Science, Vol-91, pp.823-831, 2016.

[74] J.R.S.C. Mateo, “Weighted Sum Method and Weighted Product Method”, In: Multi Criteria Analysis in the Renewable Energy Industry, Springer, London, 2012.

[75] https://in.mathworks.com/help/fuzzy/

[76] https://100xdevs.com

PUBLICATIONS

[1] Mishra, M. &Dadhich, R. (2024). "Fuzzy Logic-Based Quantification of Usability Expectation for M-Commerce Mobile Application by Using GQM and ISO 9241-11". *Journal of Computer Science*, *20*(1), 1-9. https://doi.org/10.3844/jcssp.2024.1.9 **(SCOPUS INDEXED, UGC-CARE INDEXED)**.

[2] Mishra, M., &ReenaDadhich. (2024). "A Fuzzy AHP Approach to Evaluating the Fuzzy Weight of Mobile Game Application Software Quality Factors: Usability Expectations for Novice Users". Archives of Advanced Engineering Science, 1–9.

[3] Mishra, M. &Dadhich, R. (2024). "Fuzzy Based Analysis of Software Quality Factor understand-ability for Mobile Game Application". IndianJournal of Technical Education (IJTE)**(UGC-CARE INDEXED)**.

[4] Mishra, M., Dadhich, R. (2023). "How to Quantify Software Quality Factors for Mobile Applications?: Proposed Criteria". In: Rathore, V.S., Tavares, J.M.R.S., Piuri, V., Surendiran, B. (eds) Emerging Trends in Expert Applications and Security. ICE-TEAS 2023. Lecture Notes in Networks and Systems, vol 681. Springer, Singapore.https://doi.org/10.1007/978-981-99-1909-3_27**(SCOPUS INDEXED)**.

[5] Mishra, M., Dadhich, R. (2024). "A pair-wise comparison of software quality factors for prioritization of fuzzy weight using FAHP: mobile game application usability expectations for expert user". Emerging Trends in Expert Applications and Security, ICE-TEAS 2024.**(SCOPUS INDEXED)**.

[6] Mishra, M., Dadhich, R. (2023). "Fuzzy Logic based Quantification of Usability expectation for Novice User in Mobile Games". 2nd International conference on resent trends in engineering, technology and business management (ICRTETBM 2023), Amity University, Noida **(SCOPUS INDEXED)**.

[7] Mishra, M., Dadhich, R. (2024). "Fuzzy Logic Based Usability Sub-Characteristic Analysis for M-Learning Application Utilizing GQM & ISO 9241-11". International conference on cutting edge technology in computing, communication and intelligence, ICCTCCI 2024, Cuttack (**SCOPUS INDEXED)**.

Journal of Computer Science

Original Research Paper

Fuzzy Logic-Based Quantification of Usability Expectation for M-Commerce Mobile Application by Using GQM and ISO 9241-11

Manish Mishra and Reena Dadhich

Department of Computer Science and Informatics, University of Kota, Kota, India

Article history
Received: 15-07-2023
Revised: 23-08-2023
Accepted: 27-09-2023

Corresponding Author:
Manish Mishra
Department of Computer Science and Informatics,
University of Kota, Kota, India
Email: kota.phd@gmail.com

Abstract: Fuzzy logic-based quantification of usability expectation for an m-commerce mobile application is a process of measuring the usability of a mobile application by using fuzzy logic principles. The usability of any mobile application is used to find out the user experience of the mobile application by analyzing the user's expectations and preferences. Fuzzy logic always be the optimal choice for quantification. Fuzzy logic-based quantification of usability expectation assesses the user experience of an m-commerce mobile application by taking into account the user's needs, preferences, and expectations. Usability expectation also takes into account the ability of the user to understand and interact with the application, the degree to which the application meets the user's expectations, and the overall satisfaction with the application. This process helps to identify areas of improvement, enabling the developers to make necessary changes for a better user experience. This study presents to design of a usability metric framework and then quantifies the overall usability quality of an m-commerce mobile application with the help of fuzzy logic. The proposed usability metric framework is based on the Goal-Question-Metric (GQM) approach and is intended to provide a comprehensive and systematic approach to design metrics to assess the qualitative aspect of mobile phone applications. The framework has been developed and tested in an m-commerce context and provides a set of measurable criteria to quantify m-commerce mobile applications as per standard. The results of the evaluation can then be used to improve m-commerce mobile applications and to ensure that the user experience is optimized.

Keywords: Usability, M-Commerce, GQM Approach, ISO 9241-11

Introduction

Usability for m-commerce is the degree to which users can easily, quickly and conveniently interact with an e-commerce website or mobile application. This includes factors such as navigation, layout and user interface design. Usability also includes the ability to make purchases, provide feedback, and access help quickly and easily. The goal of usability in m-commerce is to maximize user satisfaction and convenience, while also optimizing conversions. By making sure that users can quickly and easily access the information and products they need, m-commerce businesses can improve both their user experience and their bottom line.

The development of mobile and digital technologies has enabled Mobile Commerce (M-Commerce) to become an increasingly important part of the retail landscape. However, the usability of m-commerce applications remains an area of concern for industry stakeholders, particularly given the wide range of mobile devices and platforms available. As a result, there is a need for an effective usability metric framework for m-commerce applications that can be used to assess the usability of these applications across platforms. This study presents a usability metric framework for m-commerce applications using ISO 9241-11 standard. The proposed framework is designed to facilitate the assessment of m-commerce applications in terms of usability, user experience and user satisfaction. The framework is based on the ISO 9241-11 standard, which defines a set of usability metrics that can be used to evaluate the user-friendliness and effectiveness of m-commerce applications, then the proposed framework implemented as a case study of an m-commerce mobile application to illustrate the practical application of the framework. The results of the case study demonstrate the overall quantification of usability expectations of the proposed framework and provide an effective way to assess the usability of m-commerce mobile applications.

Literature Survey

Software development is always evolving. We must recognize that software development is a part of our lives and will continue to evolve. As software engineers, we must strive for excellence and build tools and ways to achieve our goals. Collaboration may enhance the future of software development. According to Basili (1989), software development activity should be considered as experimental rather than theoretical, and he presents a new project-specific model that can be better understood with the aid of knowledge, process, and product. This model is intended primarily for educational and research purposes. Excellence in software development may be accomplished by quantification based on the perspectives of many stakeholders. The establishment of a quantitative assessment approach for software engineering methods is critical for determining their efficacy. According to Basili (1985), an assessment process should contain metrics that quantify the quality of the engineering process. Thus, the goal of quantification is to improve the entire quality of software as a product. Many variables influence the overall quality of software programs. According to McCall (1977) official technical report, building high-quality software that is stable and error-free is dependent on the development of suitable metrics. These measurable indicators increase the overall quality of the Air Force system while also improving the experience of its command-and-control software. The environment in which software is produced, maintained, and utilized has an impact on its quality. As a result, it is critical to examine all of these elements while designing, maintaining, and utilizing software to ensure that it achieves the highest quality standards available. The next critical issue is if we can construct metrics in a systematic manner using an appropriate model. Yes, GQM is the correct answer. Basili (1992) GQM paradigm is a strong tool for software modeling and measurement. It offers a thorough framework for defining and assessing software objectives, questions, and measurements. This paradigm enables systematic analysis and improvement of software processes, products, and services. It may be utilized to give software stakeholders with a shared knowledge of software systems as well as an efficient platform for communication and agreement. Software engineers may use the GQM paradigm to minimize the complexity of software development while increasing the efficacy of their software models and metrics.

Software metrics are a very important tool for any firm. According to Grady and Caswell (1987), the enhancement of software development process in terms of quality, cost, and time necessitates the first steps of issue diagnosis, establishment of performance objectives, and evaluation of software project efficacy. Organizations may enhance their level of success by dedicating resources towards the establishment and execution of a comprehensive software metrics program, which enables them to get a deeper comprehension of their software development procedures.

From the perspective of each individual user, usability emerges as a vital feature. Usability refers to the evaluation of a system's effectiveness in terms of a user's ability to access, comprehend, and engage with it. According to Bevan (1995), it is essential to prioritize the consideration of quality of use as a primary design aim while developing interactive software products. This research further posits that the success of businesses is contingent upon usability, hence necessitating the establishment of usability as a primary objective in the design process. The consideration of usability is crucial in ensuring that users are able to successfully and efficiently use a system with little cognitive and physical exertion.

M-commerce, or mobile commerce, involves purchasing and selling products and services on smart phones and tablets. Today, users may use their mobile devices to make online purchases, pay bills, and manage their accounts. Mobile commerce is complicated and depends on several factors, according to Feng *et al.* (2006). Firms must understand customer needs, do thorough market research, and stay current on technology to succeed in this field. Businesses must invest in research and development, customer service, and marketing to succeed in mobile commerce.

This research introduces a novel methodology for assessing the overall quality of m-commerce apps using the GQM paradigm and ISO 9241-11 (2018) standard, using fuzzy logic as a tool.

GQM Paradigm and ISO 9241-11

The GQM paradigm has a hierarchical structure and utilizes a top-down approach. It begins by establishing goals, followed by the formulation and collection of questions, and concludes by linking metrics to each question. The process known as goal-question-metric is used to assess the effectiveness of projects and programs. This strategy facilitates the organization and analysis of information, providing guidance for decision-making and subsequent actions. The proposed approach encompasses the first step of determining a specific objective or target for a certain project or program, followed by the formulation of inquiries pertaining to the means by which it might be achieved, and then devising quantifiable measures to monitor and evaluate the advancement towards said objective.

The use of the goal-question-metric method proves to be very advantageous in scenarios when a project or program involves several stakeholders. This approach facilitates effective communication and enables the measurement of success via a shared language and framework. Additionally, this approach promotes the engagement of all relevant parties in evaluating the consequences of their choices and behaviors in relation to the overarching objective. The use of the goal-question-metric strategy enables firms to effectively identify areas for development and concentrate their endeavors on achieving the intended result.

In general, the goal-question-metric method is seen as a valuable instrument for assessing the effectiveness of a project or program. The framework offers a systematic approach for assessing results and promotes stakeholder awareness of the consequences of their choices and actions in relation to the overarching objective.

The ISO 9241-11 standard is well recognized as a benchmark for evaluating usability. This resource offers valuable insights on the use of efficient measuring techniques to enhance the usability of software products. Usability is defined as the degree to which a product may be used by designated users to successfully accomplish predetermined objectives with effectiveness, efficiency, and satisfaction within a certain context of utilization [9].

Fuzzy Logic

Fuzzy logic employs the principles of fuzzy set theory to accurately capture and articulate the intrinsic imprecision and vagueness that characterizes human cognitive processes and thinking. The fundamental idea of this notion is that problems might have several viable solutions, which requires the use of a "fuzzy" problem-solving technique to facilitate decision-making. Fuzzy logic has practical use in several domains such as image processing, robotics, and natural language processing. This particular method seems to be particularly advantageous in scenarios when conventional decision-making processes would exhibit an excessive level of rigidity or inflexibility.

Fuzzy logic is grounded on the concept of partial truth, whereby statements possess varying degrees of validity or applicability, as opposed to adhering strictly to a binary framework of absolute truth or falsehood. This approach facilitates a more intuitive decision-making process by enabling judgments to be formulated based on the available evidence, rather than only depending on a binary outcome of either a definitive positive or negative response. Fuzzy logic facilitates a heightened level of adaptability in problem-solving approaches, as it allows the system to acquire knowledge from errors and then modify its decision-making mechanism.

Table 1: Fuzzy weight and fuzzy rate

Criteria	Fuzzy weights	Fuzzy ratings
Very Low (VL)	(0.0, 0.0, 0.25)	(0.0, 0.1, 0.3)
Low (L)	(0.0, 0.25, 0.50)	(0.1, 0.3, 0.5)
Medium (M)	(0.25,0.50, 0.75)	(0.3, 0.5, 0.7)
High (H)	(0.50, 0.75, 1.0)	(0.5, 0.7, 0.9)
Very High (VH)	(0.75, 1.0, 1.0)	(0.7, 0.9, 1.0)

The use of the triangular fuzzy function methodology has been employed in the present investigation. The Triangular Fuzzy Function is a specific form of mathematical function that is often used in fuzzy logic systems for the purpose of assigning a membership value to an input value. The graph in question is a triangle form that represents the mapping of an input value to a corresponding value ranging from 0-1. In this context, 0 represents the lowest value while 1 represents the highest value. The parameters used in this investigation are derived from a publication recommended by Singh and Vidyarthi (2008), as seen in Table 1.

Fuzzy Operations

This study adopts two basic fuzzy operations explained by Ross (2004):

(a) Fuzzy multiplication
(b) Fuzzy addition

(a) Fuzzy multiplication it is defined as the pairwise product of two triangular fuzzy sets.
For example, suppose there are two triangular fuzzy sets (d, e, f) and (g, h, i) then the fuzzy multiplication is defined as:
$(d, e, f) \times (g, h, i) = (d \times g, e \times h, f \times i)$
(b) Fuzzy addition it is defined as the pairwise maximum of two triangular fuzzy sets.
For example, suppose there are two triangular fuzzy sets (d, e, f) and (g, h, i) then the fuzzy addition is defined as:
$(d, e, f) + (g, h, i) = [\max(d, g), \max(e, h), \max(f, i)]$

Fuzzification and Defuzzification

Fuzzification is defined as the process of converting a numerical input into a fuzzy set. It involves mapping a crisp input value to a fuzzy value, allowing the input to take on any value between the two. The purpose of fuzzification is to provide a more accurate representation of data by allowing greater flexibility in how inputs are interpreted.

Fuzzification is the process that converts matrices of sub-characteristics of usability into fuzzy values. Let us define the criteria by taking an example, considering the matric of error and recovery as illustrated in Table 2.

Table 2: Example of fuzzification

Metric	Description	Probability	Criteria	Fuzzy rating
Probability (P) to recover from error for each task	$P = \frac{\text{recoverable errors}}{\text{total number of errors}}$	$0<p<0.2$	VL	(0.0, 0.1, 0.3)
		$0.2<p<0.4$	L	(0.1, 0.3, 0.5)
		$0.4<p<0.6$	M	(0.3, 0.5, 0.7)
		$0.6<p<0.8$	H	(0.5, 0.7, 0.9)
		$0.8<p<1$	VH	(0.7, 0.9, 1.0)

Defuzzification is defined as the process of converting a fuzzy set of values into a single crisp value. It is the reverse of the process of fuzzification, which is the process of taking a crisp value and transforming it into a fuzzy set. This study adopts the centroid method to defuzzifuy a triangular fuzzy set:

$$\text{Centroid formula } z^* = \frac{\int \mu(z)\, z\, dz}{\int \mu(z)\, dz}$$

Here, z^* is the defuzzified crisp value, z is the value on the x-axis, and $\mu(z)$ is the membership function.

Research Approach

The Goal Question Metric (GQM) technique may help evaluate mobile phone applications. This method simplifies setting quantifiable application goals and objectives and linking them to particular metrics. Goal-oriented assessment, which sets clear, quantifiable application goals and ensures they are measured, underpins the GQM technique. GQM makes it easy to identify usability issues and provide precise usability measurements to assess their impact. The strategy requires setting user-centered and task-focused objectives. It is also crucial to choose the correct measures to assess goal achievement. The GQM approach helps quantify mobile app usefulness. Write out the application's objectives initially. After goal and target setting, the GQM approach may be utilized to analyze use precisely. Use the GQM approach to identify usability issues and create metrics to assess their effects. Compare changes over time and watch how the software evolves using GQM.

The Goal Question Metric (GQM) technique may be used to create usefulness requirements for mobile applications. The Goal-Question-Metric (GQM) technique helps create a framework with distinct objectives and questions. This framework can assess and grade mobile phone app usefulness.

To test mobile phone app usefulness using the Goal-Question-Measure (GQM) methodology, complete these steps:

1. Set a goal for the mobile app: The first step is to describe and create a goal for a mobile app for a certain area based on the requirements and standards. The goal should be clear and concise about what you want to happen when you use the app
2. Figure out the questions: The questions should be carefully thought out based on the goal of the mobile app in a certain area
3. Establish the metric: The metric should be made so that it measures how well the mobile app works with the goals and questions that have been set

Following the GQM method can help businesses create a complete framework for measuring and evaluating the usefulness of mobile phone apps. This framework will ensure a constant and dependable way to measure and evaluate the usability of mobile phone apps.

Procedure

This study proposed the following procedure to quantify the overall usability quality of m-commerce mobile applications:

Step 1: Design a Usability Metric framework that depends upon goals and questions for m-commerce mobile application

Step 2: Fuzzy weight as per developers and tester's view and fuzzy rates as per end- user's view

Step 3: Calculate of overall rating for 6 goals features, time taken, learn-ability, accuracy, security, and user's feedback by taking the user's response. Fuzzy rating of goal features, usability factor, and overall rating for usability of m-commerce mobile application calculated by the following formula:

$$\textit{Fuzzy rating} = r_1 \times w_1 + r_2 \times w_2 + \ldots\ldots\ldots\ldots + r_n \times w_n = \sum r_i \times w_i$$

Step 4: Calculation of overall fuzzy rating for 3 usability factors efficiency, effectiveness, and satisfaction

Step 5: Calculation of overall fuzzy rating for usability of m-commerce mobile application

Step 6: Calculation of crisp value of usability of the m-commerce mobile application by using the centroid method

Case Study

Suppose we are going to quantify the 'ABC' m-commerce mobile application.

DOI: 10.3844/jcssp.2024.1.9

Step 1: Design a usability metric framework that depends upon the goal and questions for 'ABC' m-commerce mobile application

Usability factor	Goal	Question	Metric
Efficiency	Features	What about display and navigation	Consistency in text, font, and colors
			Easy to navigate in mobile app
			Support night vision
		What about search pattern (text pattern and speech pattern)	Finding information about the product is easier
			Speech to text converted accurately
		What about product description	Description of each product is accurate
			Product photograph displayed adequately
			Prices of products are adequately shown
			Status (available, out of stock) of each product is adequately shown
		What about the purchase process	Easy to register
			Easy to change customer information
			Easy-to-order product
			The shopping cart's information is accurate
			Adequate information about how to order
			Adequate information about payment options
			Adequate information about how to cancel the product
			Adequate information about returns and refund policy
			Adequate information about order detail
			Adequate information about delivery time
			Adequate information about delivery cost
			Adequate information about the delivery area
			Delivery to other address
			Online order tracking is available
	Time is taken	What about time to search a sub-task	Search particular product
		What about time to complete a whole task	Time to complete task
Effectiveness	Learn-ability	What about intuitive learning about mobile app	Easy to learn interface
			Adequate content management
		What about help or customer service?	Adequate help (demo version)
			Adequate help (text version)
	Accuracy	What about mobile app response as per action	Mobile app responds properly as per action
			Every component of the interface respond accurately
			Probability of search successfully task completion in 1st attempt
		What about the chance for successful completion of task or sub-task	Probability to completion of task within given time
Satisfaction	Security	What about the security of personnel data and financial data	Adequate information about the privacy policy
			Secure socket layer used by mobile app
			Well-recognized secure payment methods
			Different modes for verification such as OTP based
	Feedback	What about the feedback process	Overall features
			Overall learning process
			Overall accuracy
			Overall security
			Overall experience using this mobile app

Step 2: Assign Fuzzy weight as per the developers and tester's view and fuzzy rates as per end- user's view

Usability factor	Goal	Metric	Average fuzzy weight	Average fuzzy rate
Efficiency	Features	Consistency in text, font, and colors	(0.5, 0.7, 0.9)	(0.7, 0.9, 1.0)
		Easy to navigate in mobile app		(0.7, 0.9, 1.0)
		Support night vision		(0.5, 0.7, 0.9)
		Finding information about the product are easier		(0.7, 0.9, 1.0)
		Speech to text converted accurately		(0.5, 0.7, 0.9)
		The description of each product is accurate		(0.5, 0.7, 0.9)
		Product photograph displayed adequately		(0.7, 0.9, 1.0)
		Prices of products are adequately shown		(0.7, 0.9, 1.0)

DOI: 10.3844/jcssp.2024.1.9

Step 2: Continue

		Status (available, out of stock) of each product is adequately shown		(0.7, 0.9, 1.0)
		Easy to register		(0.5, 0.7, 0.9)
		Easy to change customer information		(0.5, 0.7, 0.9)
		Easy to order product		(0.7, 0.9, 1.0)
		The shopping cart's information is accurate		(0.7, 0.9, 1.0)
		Adequate information about how to order		(0.5, 0.7, 0.9)
		Adequate information about payment options		(0.7, 0.9, 1.0)
		Adequate information about how to cancel the product		(0.7, 0.9, 1.0)
		Adequate information about return and refund policy		(0.7, 0.9, 1.0)
		Adequate information about order detail		(0.7, 0.9, 1.0)
		Adequate information about delivery time		(0.5, 0.7, 0.9)
		Adequate information about delivery cost		(0.7, 0.9, 1.0)
		Adequate information about the delivery area		(0.7, 0.9, 1.0)
		Delivery to other addresses		(0.7, 0.9, 1.0)
		Online order tracking available		(0.7, 0.9, 1.0)
	Time taken	Search particular product	(0.75, 1.0, 1.0)	(0.7, 0.9, 1.0)
		Time to complete a task		(0.5, 0.7, 0.9)
Effectiveness	Learn-ability	Easy to learn interface	(0.75, 1.0, 1.0)	(0.5, 0.7, 0.9)
		Adequate content management		(0.5, 0.7, 0.9)
		Adequate help (demo version)		(0.5, 0.7, 0.9)
		Adequate help (text version)		(0.7, 0.9, 1.0)
	Accuracy	Mobile app responds properly as per action	(0.75, 1.0, 1.0)	(0.7, 0.9, 1.0)
		Every component of interface responds accurately		(0.7, 0.9, 1.0)
		Probability to search successfully task completion in 1st attempt		(0.7, 0.9, 1.0)
		Probability to completion of task within given time		(0.5, 0.7, 0.9)
Satisfaction	Security	Adequate information about privacy policy	(0.75, 1.0, 1.0)	(0.7, 0.9, 1.0)
		Secure socket layer used by mobile app		(0.7, 0.9, 1.0)
		Well-recognized secure payment methods		(0.7, 0.9, 1.0)
		Different modes for verification such as OTP based		(0.7, 0.9, 1.0)
	Feedback	Overall features	(0.75, 1.0, 1.0)	(0.7, 0.9, 1.0)
		Overall learning process		(0.5, 0.7, 0.9)
		Overall accuracy		(0.5, 0.7, 0.9)
		Overall security		(0.7, 0.9, 1.0)
		Overall experience to use this mobile app		(0.5, 0.7, 0.9)

Step 3: Calculation of overall rating for 6 goals features, time taken, learn-ability, accuracy, security, and user's feedback by taking the user's response

Usability factor	Goal	Fuzzy wt	Overall fuzzy rate
Efficiency	Features	(0.75, 1.0, 1.0)	(0.35, 0.675, 1.0)
	Time taken	(0.75, 1.0, 1.0)	(0.525, 0.9, 1.0)
Effectiveness	Learn-ability	(0.75, 1.0, 1.0)	(0.35, 0.675, 1.0)
	Accuracy	(0.75, 1.0, 1.0)	(0.525, 0.9, 1.0)
Satisfaction	Security	(0.75, 1.0, 1.0)	(0.525, 0.9, 1.0)
	User's feedback	(0.75, 1.0, 1.0)	(0.525, 0.9, 1.0)

Step 4: Calculation of overall rating for 3 usability factors efficiency, effectiveness and satisfaction

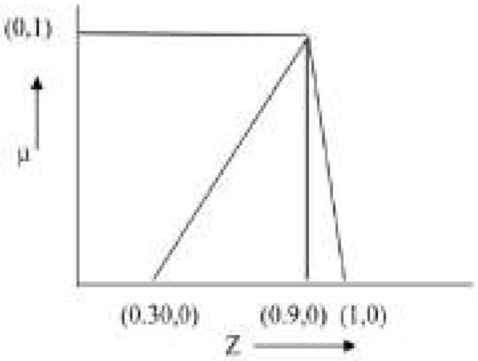
Usability factor	Fuzzy wt	Overall fuzzy rate
Efficiency	(0.75, 1.0, 1.0)	(0.394, 0.9, 1.0)
Effectiveness	(0.75, 1.0, 1.0)	(0.394, 0.9, 1.0)
Satisfaction	(0.75, 1.0, 1.0)	(0.394, 0.9, 1.0)

Step 5: Calculation of overall rating for usability of m-commerce mobile application

Overall fuzzy rating = (max $(0.394\times0.75,\ 0.394\times0.75,\ 0.394\times0.75)$, max $(0.9\times1.0,\ 0.9\times1.0,\ 0.9\times1.0)$, max $(1.0\times1.0,\ 1.0\times1.0,\ 1.0\times1.0)$)

= (0.296, 0.9, 1.0)

= (0.30, 0.9, 1.0)



Step 6: Calculation of crisp value of usability of the m-commerce mobile application by using the centroid method

Equation of line passing through (0.30, 0) and (0.9, 1):

$$\mu = 1.67\,z - 0.5$$

Equation of line passing through (1, 0) and (0.9, 1):

$$\mu = 10 - 10z$$

$$Z^* = \frac{\int_{0.3}^{0.9}(1.67\,z - 0.5)z\,dz + \int_{0.9}^{1}(10 - 10\,z)z\,dz}{\int_{0.3}^{0.9}(1.67\,z - 0.5)dz + \int_{0.9}^{1}(10 - 10\,z)dz}$$

On evaluating the above integral obtained value is 0.7422, thus the crisp value as per centroid method = 0.7422 or 74.22%. Thus, the overall quality of ABC m-commerce mobile application will be calculated as 74.22%.

Materials and Methods

This paper proposed a fuzzy based mathematical model which converts qualitative aspect of usability for m-commerce application in to quantitative aspect. Quantitative aspect always is a best way for any kind of analysis. This paper builds a structured framework for usability metrics based upon GQM and ISO-9241-11. Hence this framework accepts corresponding weight and rate for each matrices based on fuzzy parameter implemented by triangular fuzzy set. There are three usability factors efficiency, effectiveness and satisfaction, all these usability factors are computed with the help of fuzzy weight and fuzzy rate of matrices. The overall fuzzy rating for usability computed with the help of usability factors efficiency, effectiveness and satisfaction. The mathematical method involves as derivation of equation of two straight lines for resultant triangular fuzzy value as:

$$\mu = 1.67\,z - 0.5 \qquad (1)$$

$$\mu = 10 - 10z \qquad (2)$$

The overall fuzzy value then converted in to corresponding crisp value with the help of centroid method for defuzzification as:

$$Z^* = \frac{\int_{0.3}^{0.9}(1.67\,z - 0.5)z\,dz + \int_{0.9}^{1}(10 - 10\,z)z\,dz}{\int_{0.3}^{0.9}(1.67\,z - 0.5)dz + \int_{0.9}^{1}(10 - 10\,z)dz}$$

Which evaluates overall crisp rating or overall quality equivalent to overall qualitative aspect.

Results

The usability framework evaluates overall fuzzy rating with the help of corresponding fuzzy weights and fuzzy rating as (0.30, 0.9, 1.0). This study proposes theoretical mathematical framework which derives two lines of equation one passing through (0.30, 0) and (0.9, 1) derived as:

$$\mu - 0 = \frac{(1.0-0)}{(0.9-0.3)}\,(z - 0.30)$$

$$\mu = \frac{(1.0)}{(0.6)}\,(z - 0.30)$$

$$\mu = \left(\frac{z}{0.6} - \frac{0.3}{0.6}\right)$$

$$\mu = 1.67\,z - 0.5 \qquad (1)$$

and other passing through (1, 0) and (0.9, 1) derived as:

$$\mu - 0 = \frac{(1.0-0)}{(0.9-1.0)}\,(z - 1.0)$$

$$\mu = \frac{(1.0)}{(-0.1)}\,(z - 1.0)$$

$$\mu = \left(\frac{z}{-0.1} + \frac{1.0}{0.1}\right)$$

$$\mu = 10 - 10z \qquad (2)$$

Computation performed by defuzzification as:

$$Z^* = \frac{\int_{0.3}^{0.9}(1.67\,z - 0.5)z\,dz + \int_{0.9}^{1}(10 - 10\,z)z\,dz}{\int_{0.3}^{0.9}(1.67\,z - 0.5)dz + \int_{0.9}^{1}(10 - 10\,z)dz}$$

$$Z^* = \frac{0.211 + 0.05}{0.301 + 0.05}$$

$$Z^* = \frac{0.261}{0.351}$$

$$Z^* = 0.7436 \text{ or } 74.36\%$$

Thus the overall quality will be evaluated as 74.36%.

Discussion

The utilization of quantitative data guarantees the objectivity of decisions. It is useful for determining which areas require improvement, ranking the importance of additions or updates, and enhancing the development process's efficiency. With this, the software development company will be able to establish precise and quantifiable success metrics. This enables the establishment of measurable objectives and the tracking of progress over time. Quantitative data may also help allocate available resources more efficiently. If you measure the impact of the various additions or modifications you implement, you can focus your efforts on the areas that provide the highest return on investment. This strategy will provide a solid foundation for the successful conversion of qualitative aspects to quantitative evaluation. Therefore, it is recommended to all those enterprises, who produce mobile applications, that this framework approaches towards adequate computation. This will guarantee that app development organizations are able to give a comprehensive knowledge of their app's performance and user happiness via reliable quantitative data. As a result, it will be possible to enhance trust among investors and stakeholders.

This study presents a robust and innovative approach to assess the comprehensive quantitative software quality standards of M-commerce apps, focusing on the viewpoint of end users. This methodology supports stakeholders within mobile application enterprises in their pursuit of accurate computation for the analysis of overall quality throughout the development of mobile applications.

Conclusion

This research study has presented quantification of usability expectations for m-commerce mobile applications with the help of ISO 9241-11, the GQM approach, and fuzzy logic. It has demonstrated that fuzzy logic is a viable tool for quantifying usability expectations and providing an effective framework to explore the usability goals of a mobile application. The results obtained from the proposed approach are promising in terms of providing a comprehensive and accurate evaluation of the usability of m-commerce mobile applications. In terms of future scope of research, further studies could be conducted to evaluate the effectiveness of the proposed approach in different application domains. Furthermore, a comparative study could be conducted to explore the relative performance of fuzzy logic-based quantification of usability expectations for m-commerce mobile applications when compared to other approaches. Moreover, a study could be conducted to explore the potential of fuzzy logic-based quantification of usability expectations in the context of mobile application security. Overall, this research paper has provided a comprehensive framework for quantifying the usability expectation of m-commerce mobile applications with the help of fuzzy logic. The proposed approach shows potential for use in various application domains and offers a promising future for the development of mobile applications.

Acknowledgment

I express my gratitude to Prof. Reena Dadhich for her invaluable motivational support and guidance in the composition of this article.

Funding Information

The authors did not receive any monetary assistance for the investigation, composition, and/or publication of this manuscript.

Author's Contributions

Manish Mishra: Contributed to the conceptualization, formal analysis, methodology section, coding, validation, and original drafted written phases of the study.

Reena Dadhich: Contributed to the conceptualization, supervision, reviewed, edited and administration of the project.

Ethics

The work accurately and thoroughly represents the authors' research and analysis. The work acknowledges the valuable contributions of coauthors and co-researchers. The findings are properly put in research history. Text cites all references and related works.

Conflict of Interest

The authors hereby declare their complete independence from any organization or entity that may have a financial or non-financial interest in the subject matter or materials discussed in this manuscript.

References

Basili, V. R. (1985). Quantitative evaluation of software methodology. https://www.cs.umd.edu/~basili/publications/proceedings/P29.pdf

Basili, V. R. (1989, September). Software development: A paradigm for the future. In *[1989] Proceedings of the 13th Annual International Computer Software & Applications Conference* (pp. 471-485). IEEE. https://doi.org/10.1109/CMPSAC.1989.65127

Basili, V. R. (1992). Software modeling and measurement: The Goal/Question/Metric paradigm. https://www.cs.umd.edu/users/mvz/handouts/gqm.pdf

Bevan, N. (1995). Usability is quality of use. In *Advances in Human Factors/Ergonomics* (Vol. 20, pp: 349-354). Elsevier. https://doi.org/10.1016/S0921-2647(06)80241-8

Feng, H., Hoegler, T., & Stucky, W. (2006, June). Exploring the critical success factors for mobile commerce. In *2006 International Conference on Mobile Business* (pp. 40-40). IEEE. https://doi.org/10.1109/ICMB.2006.15

Grady, R. B., & Caswell, D. L. (1987). *Software metrics: Establishing a company-wide program*. Prentice-Hall, In. https://dl.acm.org/doi/abs/10.5555/26775

ISO 9241-11. (2018). Ergonomics of human-system interaction-Part 11: Usability: Definitions and concepts. https://www.iso.org/standard/63500.html

McCall, J. A. (1977). Factors in software quality. *US Rome Air Development Center Reports*. https://cir.nii.ac.jp/crid/1572261549355424000

Ross, T. J. (2004). Fuzzy Logic with Engineering Applications, 2nd Ed., Wiley India Pvt. Ltd, New Delhi, India. ISBN:10-8126513573.

Singh, A. P., & Vidyarthi, A. K. (2008). Optimal allocation of landfill disposal site: A fuzzy multi-criteria approach. https://tspace.library.utoronto.ca/bitstream/1807/62113/1/se08004.pdf

Received: 30 October 2023 | Revised: 5 February 2024 | Accepted: 20 March 2024 | Published online: 26 March 2024

RESEARCH ARTICLE

Archives of Advanced Engineering Science
2024, Vol. 00(00) 1–9
DOI: 10.47852/bonviewAAES42021974

A Fuzzy AHP Approach to Evaluating the Fuzzy Weight of Mobile Game Application Software Quality Factors: Usability Expectations for Novice Users

Manish Mishra[1,*] **and Reena Dadhich**[1]

[1]*Department of Computer Science & Informatics, University of Kota, India*

Abstract: Mobile games have become a popular way to pass the time, and an increasing number of new and inexperienced users are searching for enjoyable and simple-to-learn experiences. It is essential to ensure that the software of these mobile games is intuitive and of high quality in order to keep users satisfied and coming back. However, it is difficult to assess the subjective and frequently nebulous expectations of first-time users. In between the development of mobile game applications, stakeholders in the mobile game industry are continually exploring applications whose quality is measured. The deployment of this effort will be facilitated by the quantification of qualitative aspects that depend on software quality factors. Thus, it is crucial to choose the proper evaluation strategy. The evaluation is based on two parameters: fuzzy rate and fuzzy weight. The evaluation of a five-point fuzzy rating system is based on surveys or questionnaires, and it involves straightforward inputs. Fuzzy weight refers to the relative significance of software quality aspects, and it is determined by stakeholders responsible to ensure appropriate quality of mobile application, ensuring that the crisp sum is equal to one. This may be achieved when stakeholders agree on the relative significance or pairwise comparison of software quality factors and examine their relative value. This technique guarantees that the total assessment of quality converges in the correct direction. This paper presents a novel application of the fuzzy analytic hierarchy process that evaluates the appropriate fuzzy weight for software quality factors of mobile game applications, particularly for neophyte users. This will aid the mobile game application industry in formulating and evaluating fuzzy weights to reinforce the process that converts qualitative to quantitative aspects during mobile application development.

Keywords: software quality factors, mobile game, quantification, fuzzy logic, usability, novice users, FAHP

1. Introduction

Getting software that is both easy to use and of good quality is always hard in the world of mobile games, where user experience is everything and things change quickly. How well these things work together determines how well mobile games do, especially with people who have never played them before or are new to using their phones. New gamers want games that are more fun and easy to use as the industry grows. Mobile game developers have to change every part of their games and make them better than their competitors in order to meet these standards and do better than their competitors.

The quality of the mobile application as a final product is the only thing that will be able to do this, but there are two essential questions that need to be answered: what are the quality factors? Moreover, what are the criteria that are used to assess quality factors throughout the creation of the application? A method to fuzzy-based assessment is presented in this study. The evaluation model is dependent on two parameters: fuzzy rating and fuzzy weight. The survey or questionnaire that is based on the prototype of the mobile application that is currently being developed is the source of the input for fuzzy ratings on a five-scale fuzzy framework, such as VL, L, M, H, and VH. At the same time, the input source of fuzzy weight may also be taken as five scale fuzzy weights, such as VL, L, M, H, and VH; nevertheless, weight is subject to one significant restriction, which is that the total must equal one. The current literature does not take this into account. The purpose of this study is to address the research gap that exists between the current literature and assessment criteria that potential mobile application companies may use in order to evaluate the quality of applications while they are being developed and access the overall quality in the appropriate direction. In order to achieve the research objective of this study, it is essential to accurately evaluate the quality of the application or quantify its overall quality throughout the development process of a mobile application.

An implicit feature that is of tremendous value for any assessment is the ability to perform precise calculations or accurate measurements. The emphasis of this research is on this particular feature, which contributes to more accurate assessment and, as a result, converges to an adequate level of overall quality. Every time a mobile application is developed by any software or mobile application, stakeholders or investors always place a

*__Corresponding author:__ Manish Mishra, Department of Computer Science & Informatics, University of Kota, India. Email: profrdadhich@uok.ac.in

strong emphasis on the quality of the mobile application while it is being developed. Due to the fact that quality is dependent on quality factors, the first thing that has to be done is to gather the ideal amount of quality factors according to the stakeholders, investors, and quality managers or decision makers of mobile application development companies. The next phase in the process involves quantifying the overall quality of the mobile application by considering the optimal combination of quality factors that require input for fuzzy rate and fuzzy weight. This research study focuses on the notion that the cumulative weight of quality factors must be precisely equal to one. The purpose of this component is to ensure the accuracy of fuzzy weights in all calculations related to overall quality. It also aids decision makers in correctly interpreting these calculations during the development of mobile applications.

The fuzzy analytic hierarchy process (FAHP) is the subject of this study paper. It is a new and cutting-edge way to figure out how to measure value expectations in the context of software quality factors for mobile game newbies. Traditional ways of testing usefulness do not always show how new users' needs are different and often subjective. This is especially true now that mobile game apps are always getting more involved and different.

In our study, we look into fuzzy weights and come up with a new framework that takes into account the questions and contradictions that happen naturally when judging the expectations of mobile gamers. The fuzzy AHP was added so that we can meet the needs of both people who use AHP in simple, clear ways and people who need to use it in more complicated, specific ways. This method is meant to give more accurate and adaptable details on how to select the most important fuzzy weight software quality factors. A lot of people will be able to understand and use the power of software quality factors if they know how to properly evaluate the fuzzy weight of quality factors.

It is important to remember that meeting standards for software quality and value is a hard problem that needs advanced methods to solve as mobile games change the way people have fun. In this way, this paper can be used as a guide for further research and as a valuable tool for game developers who want to improve the way they evaluate the quality of games and make mobile games that keep getting more and more people excited.

The purpose of this article is to establish the appropriate framework, together with the ideal amount of quality factors, for the purpose of making mobile games simple to learn for first-time players. On the other hand, it makes use of fuzzy AHP in order to determine the relative importance of each software quality factor (QF) for mobile gaming apps while they are in the process of being developed.

2. Literature Review

The term "quality" is the one that is often used in describing any form of product. Software is anything that may be considered a product; hence, it is unclear how someone can argue that software is of high quality? Numerous professionals have devoted their time and energy to researching and developing straightforward models to describe quality with the help of quality models, such as Boehm's model [1] and Dromey's model [2]. A framework for assessing software quality and developing a quality model was proposed by Maryol et al. [24]. The International Organization for Standardization (ISO) is a global organization that establishes several quality standards. One of these is the ISO 9126 standard [3]. However, each of these models is generic in nature, but they may be modified to meet the requirements of any software application, taking into account the constraints of the particular software application. These concepts offer a foundation for excellence as a foundation. The quality framework is a qualitative element of the software product, and this one is useless if we do not have a comparison analysis for the quality aspects or a technique to figure out the total quality in a numerical approach.

The use of fuzzy logic is required for any kind of quantitative or qualitative analysis. Since the evaluation process contains ambiguity, fuzzy logic is one of the greatest techniques for dealing with such situations [4].

Thus, the next important question is how to quantify the quantity attributes. The quantification of quality factors investigates a method to examine quality variables or justify overall quality while the software product is being developed. Many of the researchers are moving in this particular direction. In their study, Srivastava et al. [5] examine the application of a fuzzy multicriteria approach to quantify software quality. A method for quantifying the quality of software is proposed by Dubey and Sharma [6] using a multicriteria decision approach. Similarly, Reena and Bhavesh [7] suggest a technique for assessing the dependability of aspect-oriented software by employing fuzzy logic. Srivastava and Kumar [8] put forth the perspectives of a project manager, a developer, and a tester, and further developed the research. However, the focus of these studies remains on developing a framework to ensure the production of high-quality software products and quantifying them as desktop applications. The next significant step in the process is to transform qualitative aspects into quantitative aspects for a mobile application [9]. Nitze and Schmietendorf [10] conduct a survey to assess the perception and expectation of mobile consumers regarding software quality. Idri et al. [11] modify the ISO 9126 standard to accommodate mobile environments.

The input of the end user is critical to the determination of one of the most crucial aspects of product quality, i.e., usability. Bevan-defined usability as quality of use [12]. The issue of usability arises as a result of consumers' responses to the product, which might be positive or negative communication among customers in the future. Similar to how Ghazizadeh and Vafadar [13] concentrated on quantitative usability evaluation, this article also proposes an empirical investigation. Mobile gaming is one of the most complex domains, where considerable effort was required to ensure that mobile game applications were usable. Cui and Zhu [14] present a model for determining the most effective user interface from a variety of alternatives by analyzing user interviews and questionnaire responses and satisfying user expectations. The ease of use of a mobile gaming application is and will continue to be a key factor, particularly for first-time users and other inexperienced players. In order to have a good usability quality framework, one must ensure that the optimal amount of quality elements is present [15]. A framework for assessing the efficacy of first-time user experience (FTUE) in mobile games was introduced by Barnett et al. [16].

The evaluation procedure is determined by the relative importance of the many quality parameters. The first step toward a more effective assessment procedure will be determining the ideal value of the weights. AHP, which stands for "analytic hierarchy process," is one of the ways in which pairwise comparison is conducted to determine the relative significance or preference of items by comparing them in pairs. This is accomplished by giving numerical numbers that signify the intensity of preference. These values are often represented on a scale, such as the Saaty scale, where a value of 1 implies equal significance, but other values imply different levels of preference.

This approach helps where anyone may determine an acceptable weight by doing pair-by-pair comparisons, introduced by Saaty [17].

The FAHP, often known as FAHP, was first presented by Buckley in 1985 as an extension of the AHP [18]. Fuzzy AHP enhances the conventional AHP methodology by using fuzzy logic. Fuzzy logic enables the incorporation of ambiguity and inexactness in the process of decision making. Fuzzy AHP employs linguistic variables and fuzzy numbers to indicate the ambiguity linked to judgments, as opposed to using precise values. In a FAHP, the pairwise comparison matrix is expanded to include fuzzy numbers. Instead of giving an exact value, decision makers have the option to articulate their judgments using language phrases such as "slightly more important" or "strongly more important," which are then translated into fuzzy numbers.

The FAHP may be used in a wide variety of different contexts (Wu, Tzeng & Chen, 19).

Data envelopment analysis (DEA) is another quantitative technique used in the fields of operations research and management science to evaluate the effectiveness of decision-making processes. The research of Sharafi et al. [20] introduces a novel fuzzy DEA model for the purpose of selecting green suppliers. The model does this by gathering expert votes. Subsequently, the suggested models were used to choose an environmentally friendly supplier inside an automobile conglomerate. In this particular case study, a comprehensive rating of environmentally friendly providers is achieved.

Decision makers may handle uncertainty and imprecision in the decision-making process by combining fuzzy AHP with pairwise comparison. A novel approach proposed by Tavana et al. [21] to enhance the limitations of DEA by incorporating the advantages of pairwise comparisons in AHP. It also proposes many new hybrid MADM-DEA models with varying levels of computing complexity and consistency.

The determination of fuzzy weights is the primary topic of discussion in this chapter [22], with a specific emphasis placed on the fuzzy least square error approach and the fuzzy best worst approach. The purpose of this chapter is to provide readers with a better knowledge of fuzzy weight determination techniques and the possible applications of these approaches in decision-making situations that occur in the real world.

Pairwise comparison is very advantageous in complex decision-making situations that include numerous factors. It facilitates the process of making comparisons and offers a systematic method for assessing different choices. The analytic hierarchy process (AHP) is a frequently used technique for pairwise comparison. It entails the creation of a matrix to systematically assess and evaluate alternates.

This article describes a one-of-a-kind application with the support of FAHP and pairwise comparison with the goal of defining fuzzy weight of software quality factors for mobile gaming applications according to the usability expectations of novice users.

3. Fuzzy Logic

Fuzzy logic is a way of thinking about mathematics as well as a kind of logic that supports persons in forming judgments and reasoning when they are uncertain about what the future holds. Fuzzy logic is also a type of logic. In contrast to classical logic, often known as Boolean logic, which can only accept "true" or "false" values as inputs, fuzzy logic is able to take in values that are neither true nor false. It is willing to work with values in the corner that represent varying degrees of truth or positions within a collection. Because of this, it is useful when working with information that is not crystal clear or precise. When it comes to the concept of variables, linguistic variables have numbers that are represented by language words or concepts, such as "low," "medium," and "high." These expressions are not used to discuss specific numerical values, but rather broad or qualitative aspects of a system. You may, for instance, have a linguistic variable known as "temperature" that contains terms such as "cold," "cool," "warm," and "hot" that together describe the temperature of a particular location. When there is a great deal of uncertainty and potential for making mistakes, fuzzy logic and linguistic elements are often utilized jointly to make judgments because of the tight connection between the two. This paper adopts a triangular fuzzy membership function as illustrated in Figure 1.

Figure 1
Triangular fuzzy membership function

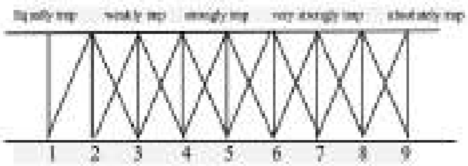


Table 1
Relationship between linguistic variable and fuzzy value as per Figure 1

Comparison (crisp)	Linguistic variable	Fuzzy value
1	Equally important	(1,1,1)
2	In between equally important and weakly importance	(1,2,3)
3	Weakly importance	(2,3,4)
4	In between weakly important and strongly importance	(3,4,5)
5	Strongly importance	(4,5,6)
6	In between strongly important and very strongly importance	(5,6,7)
7	Very strongly important	(6,7,8)
8	In between very strongly important and absolutely importance	(7,8,9)
9	Absolutely important	(8,9,9)

Table 2
Relationship between crisp value & inverse fuzzy value

Comparison (crisp)	Fuzzy value (Inverse)	Simplified fuzzy value
1/1	$(1,1,1)^{-1}$	$(1,1,1)$
1/2	$(1,2,3)^{-1}$	$(\frac{1}{3},\frac{1}{2},\frac{1}{1})$
1/3	$(2,3,4)^{-1}$	$(\frac{1}{4},\frac{1}{3},\frac{1}{2})$
1/4	$(3,4,5)^{-1}$	$(\frac{1}{5},\frac{1}{4},\frac{1}{3})$
1/5	$(4,5,6)^{-1}$	$(\frac{1}{6},\frac{1}{5},\frac{1}{4})$
1/6	$(5,6,7)^{-1}$	$(\frac{1}{7},\frac{1}{6},\frac{1}{5})$
1/7	$(6,7,8)^{-1}$	$(\frac{1}{8},\frac{1}{7},\frac{1}{6})$
1/8	$(7,8,9)^{-1}$	$(\frac{1}{9},\frac{1}{8},\frac{1}{7})$
1/9	$(8,9,9)^{-1}$	$(\frac{1}{9},\frac{1}{9},\frac{1}{8})$

4. Fuzzy AHP

The regular AHP is enhanced with the FAHP, which adds fuzzy reasoning to the normal AHP in order to assist individuals in making choices when they are at a loss for what to do. AHP stands for "analytical hierarchy process," and it is a method for making decisions that guide you through looking at a list of possibilities and ranking them based on a variety of criteria. Thomas L. Saaty was the one who first came up with the concept in the 1970s. The AHP is expanded upon by the fuzzy AHP method, which enables users to make choices based on less-than-perfect conclusions and unknowns. People who make judgments in the real world do not always have crystal clear and accurate figures to compare, and there could be some ambiguity. Words and phrases such as "very important," "somewhat important," and "not important" are employed in fuzzy logic to demonstrate these confusing conclusions.

5. Research Methodology

5.1. Research design

A model of evaluation that is fuzzy based is shown in this study. In order to do this, we use ISO 9126 as our point of departure. Learn-ability, attractiveness, and operability are the three quality factors that need to be extracted from ISO 9126. Simplicity, error and recovery, and user satisfaction are the other three qualities that have to be taken into consideration. Therefore, our proposed model consists of a total of six distinct quality factors as illustrated in Table 1.

This research adopts the strategy that is presented to do a pair-wise comparison among all of the quality parameters, the justification for this method is that first to find out the relative significance between each of them being taken into consideration. Therefore, this study creates a pair-wise matrix; but, prior to this, it is important to determine the relative value of each quality element when seen in a pair-wise fashion as illustrated in Table 2. Uncertainty surrounds the nature of weight. The triangular fuzzy function is considered in this study.

Table 3
Quality factors to access usability

Characteristic	Quality factors
Usability	Learn-ability Simplicity Attractiveness Operability Error and recovery User satisfaction

5.2. Proposed model

This paper adopts the algorithm discussed by Nidhhan et al. ([23], P-826). The following steps will explain the proposed model.

Step 1. Design the usability framework for mobile game applications.
Step 2. Decide pair-wise priorities (crisp value) among software quality factors of usability.
Step 3. Replace crisp value with corresponding fuzzy values as per linguistic variable.
Step 4. Calculate fuzzy geometric mean (FGM) for all quality factors (QF).
Step 5. Calculate normalized fuzzy weight (NFW) from FGM for all quality factors (QF).
Step 6. Convert NFW into corresponding crisp values and validate with the sum of all quality factors (QF).

5.3. Data analysis and validation

This study uses five criteria as linguistic variables: equally imp, weakly imp, strongly imp, very strongly imp, and absolutely imp. Each of these criteria maps to a fuzzy value according to the triangle fuzzy function. The data source for fuzzy weight is determined by assessing the degree of relative significance among all quality factors in a pair-wise manner, as described in step 1 of Section 6 (Case Study).

The fuzzy value of the data that has to be acquired, which is reliant on the data values, is shown in Table 2. These data were organized in the form of a pairwise matrix so that each pair of QF could be compared to one another in a pairwise way. The validation of the data is shown by the fact that this matrix converges to a normalized pair-wise matrix, in which the sum of the crisp sums of each QF is considered to be one.

6. Case Study

Suppose there is a mobile development organization "ABC" whose stakeholders evaluate fuzzy weight for six quality factors QF1 to QF6. The following steps along with numeric values illustrate it. Quality factors for usability expectation as per novice users for mobile game applications are illustrated in Table 3:

Table 4
Quality factors QF1 to QF6

	Quality factors
QF1	Learn-ability
QF2	Simplicity
QF3	Attractiveness
QF4	Operability
QF5	Error and recovery
QF6	User satisfaction

Step 1: Design a pair-wise comparison framework matrix for each pair of quality factors. Stakeholders and investors mutually agreed upon the following criteria imposed upon six quality factors as illustrated in Table 4:

1. Learn-ability is 4 times more important (in between weakly imp and strongly imp) than simplicity.
2. Learn-ability is 7 times more important (very strongly imp) than attractiveness.
3. Learn-ability is 6 times more important (in between weakly imp and strongly imp) than error and recovery.
4. Learn-ability is 3 times more important (weakly imp) than operability.
5. Learn-ability is 8 times more important (in between v strong imp and absolutely imp) than user satisfaction.
6. Simplicity is 5 times more important (strongly imp) than attractiveness.
7. Simplicity is 3 times more important (weakly imp) than error and recovery.
8. Simplicity is 7 times more important (very strongly imp) than user satisfaction.
9. Operability is 2 times more important (in between equally imp and weakly imp) than simplicity.
10. Error and recovery are 2 times more important (in between equally imp and weakly imp) than attractiveness.
11. Operability is 6 times more important (in between strongly imp and very strongly imp) than attractiveness.
12. Attractiveness is 5 times more important (strongly imp) than user satisfaction.
13. Operability is 4 times more important (in between weakly imp and strongly imp) than error and recovery.
14. Error and recovery is 5 times more important (strongly imp) than user satisfaction.
15. Operability is 7 times more important (very strongly imp) than user satisfaction.

Table 5
Pair-wise comparison matrix

	QF1	QF2	QF3	QF4	QF5	QF6
QF1	1	4	7	6	3	8
QF2	1/4	1	5	3	1/2	7
QF3	1/7	1/5	1	1/2	1/6	3
QF4	1/6	1/3	2	1	1/4	5
QF5	1/3	2	6	4	1	7
QF6	1/8	1/7	1/3	1/5	1/7	1

Step 2. Replace fuzzy value with crisp values as per Tables 2 and 5 and calculate FGM for each QF as illustrated in Table 6.

Computation of FGM for quality factor QF1:

$$\left((1\times3\times6\times5\times2\times7)^{1/6}, (1\times4\times7\times6\times3\times8)^{1/6}, (1\times5\times8\times7\times4\times9)^{1/6}\right) = (3.37, 4.10, 4.79)$$

Similarly calculated for other quality factors.

Step 3. Computation of NFW for each quality factor as illustrated in Table 7.

Computation of NFW for quality factor QF1:

$$(3.37, 4.10, 4.79) * \left(\frac{1}{11.36}, \frac{1}{9.25}, \frac{1}{7.37}\right) = (0.297, 0.443, 0.649)$$

Similarly calculated for other quality factors.

Table 7
Computation of normalized fuzzy weight (NFW) for each quality factor

	FGM	NFW
QF1	(3.37,4.10,4.79)	(0.297,0.443,0.649)
QF2	(1.22,1.55,2.03)	(0.107,0.168,0.275)
QF3	(0.34,0.43,0.56)	(0.029,0.046,0.076)
QF4	(0.55,0.71,0.92)	(0.048,0.077,0.125)
QF5	(1.69,2.23,2.79)	(0.149,0.241,0.379)
QF6	(0.2,0.23,0.27)	(0.018,0.025,0.037)

Step 4. Computation of crisp weight for each quality factor and verify with its sum which is 1.06 (nearly equal to 1). Hence, NFW for each quality factor can be used for computation of usability expectations of mobile games for novice users. Crisp weight also justifies overall relative importance in between quality factors as illustrated in Table 8.

Thus, the mobile app development company "ABC" has a mathematically sound method of calculating fuzzy weights.

Table 8
Verification with crisp weight (Sum = 1) and priorities of quality factors

	NFW	Crisp weight	Priority
QF1	(0.297,0.443,0.649)	0.463	1st
QF2	(0.107,0.168,0.275)	0.183	3rd
QF3	(0.029,0.046,0.076)	0.050	5th
QF4	(0.048,0.077,0.125)	0.083	4th
QF5	(0.149,0.241,0.379)	0.256	2nd
QF6	(0.018,0.025,0.037)	0.027	6th

Table 6
Computation of fuzzy geometric mean (FGM) for each quality factor

	QF1	QF2	QF3	QF4	QF5	QF6	FGM
QF1	(1,1,1)	(3,4,5)	(6,7,8)	(5,6,7)	(2,3,4)	(7,8,9)	(3.37,4.10,4.79)
QF2	(1/5,1/4,1/3)	(1,1,1)	(4,5,6)	(2,3,4)	(1/3,1/2,1)	(6,7,8)	(1.22,1.55,2.03)
QF3	(1/8,1/7,1/6)	(1/6,1/5,1/4)	(1,1,1)	(1/3,1/2,1)	(1/7,1/6,1/5)	(2,3,4)	(0.34,0.43,0.56)
QF4	(1/7,1/6,1/5)	(1/4,1/3,1/2)	(1,2,3)	(1,1,1)	(1/5,1/4,1/3)	(4,5,6)	(0.55,0.71,0.92)
QF5	(1/4,1/3,1/2)	(1,2,3)	(5,6,7)	(3,4,5)	(1,1,1)	(6,7,8)	(1.69,2.23,2.79)
QF6	(1/9,1/8,1/7)	(1/8,1/7,1/6)	(1/4,1/3,1/2)	(1/6,1/5,1/4)	(1/8,1/7,1/6)	(1,1,1)	(0.2,0.23,0.27)

6.1. Results and discussion

The results that were acquired inside this part have two points of view. The first point is that the total of the crisp weight of all quality factors equals 1.06, which is considered to be one, and this is what validates the data. The other is the order of priority among the quality factors in comparison to the crisp weight, which may be placed in accordance with the order of priority as shown in Table 8.

The results that have been validated guarantee that there is a significant link between the mathematical theoretical framework and the results that have been seen. The results that have been validated additionally guarantee that there is a significant association between the theoretical framework and its practical application.

6.2. A practical approach

The result that was acquired in this study was applied by the decision makers as an allocation of fuzzy weight, as illustrated in Table 9. The information presented here offers a concept of the distribution of fuzzy weights on a scale of five, according to the decision maker, and one feasible combination is illustrated in Table 10.

Table 9
Relative distance in six quality factors

	Relative distance
QF1-QF5	0.207
QF5-QF2	0.073
QF2-QF4	0.1
QF4-QF3	0.033
QF3-QF6	0.023

Table 10
Allocation of five-scale fuzzy weight as per relative distance by decision maker of the mobile industry

	Crisp weight	Priority	Fuzzy weight
QF1	0.463	1st	VH
QF2	0.183	3rd	H
QF3	0.050	5th	M
QF4	0.083	4th	M
QF5	0.256	2nd	H
QF6	0.027	6th	M

This paper adopts five-scale of triangular fuzzy weight and triangular fuzzy rate as illustrated in Table 11.

Table 11
Fuzzy weight and fuzzy rate

Criteria	Fuzzy weight	Fuzzy rating
VL (Very low)	(0.0,0.0,0.25)	(0.0,0.1,0.3)
L (Low)	(0.0,0.25,0.50)	(0.1,0.3,0.5)
M (Medium)	(0.25,0.50,0.75)	(0.3,0.5,0.7)
H (High)	(0.50,0.75,1.0)	(0.5,0.7,0.9)
VH (Very high)	(0.75,1.0,1.0)	(0.7,0.9,1.0)

Evaluation of fuzzy rating as feedback obtained by end user. One of the possible outcomes is illustrated in Table 12.

Table 12
Allocation of five-scale fuzzy weight as per relative distance by decision maker of the mobile industry

	Fuzzy weight	Fuzzy rate
QF1	(0.75,1.0,1.0)	(0.7,0.9,1.0)
QF2	(0.50,0.75,1.0)	(0.5,0.7,0.9)
QF3	(0.25,0.50,0.75)	(0.7,0.9,1.0)
QF4	(0.25,0.50,0.75)	(0.5,0.7,0.9)
QF5	(0.50,0.75,1.0)	(0.7,0.9,1.0)
QF6	(0.25,0.50,0.75)	(0.5,0.7,0.9)

The overall fuzzy rating of overall quality (fuzzy) is obtained with the help of fuzzy multiplication and fuzzy addition is illustrated in Table 13.

Table 13
Evaluation of overall fuzzy rating or overall quality in fuzzy term as per novice users

	Fuzzy weight	Fuzzy rate	Overall rating
QF1	(0.75,1.0,1.0)	(0.7,0.9,1.0)	(0.525,0.9,1.0)
QF2	(0.50,0.75,1.0)	(0.5,0.7,0.9)	(0.25,0.525,0.9)
QF3	(0.25,0.50,0.75)	(0.7,0.9,1.0)	(0.175,0.45,0.75)
QF4	(0.25,0.50,0.75)	(0.5,0.7,0.9)	(0.125,0.35,0.675)
QF5	(0.50,0.75,1.0)	(0.7,0.9,1.0)	(0.35,0.675,1.0)
QF6	(0.25,0.50,0.75)	(0.5,0.7,0.9)	(0.125,0.35,0.675)

Overall fuzzy rating obtained as (0.525, 0.9, 1.0), when defuzzify with the help of the centroid method, overall crisp quality obtained as 0.8083 or 80.83%.

6.3. MATLAB simulation

MATLAB was used to provide a visual representation of the overall quality. Two of the inputs are fuzzy weight and fuzzy rate, both of which are implemented by triangle fuzzy membership and are depicted in Figures 2 and 3, respectively. The output function is an overall fuzzy rating for beginner users, which is implemented as a triangular fuzzy membership function and is illustrated in Figure 4. Figure 5 is an illustration of the control surface graph in MATLAB, which was used to verify the data acquired, which showed that the overall crisp quality was attained as 0.8024, which is equivalent to 80.24%. The purpose of this practical example is to demonstrate how the priority of fuzzy weight plays a significant part in an effort to attain overall correct quality.

Figure 2
Triangular fuzzy membership function (fuzzy weight) as input

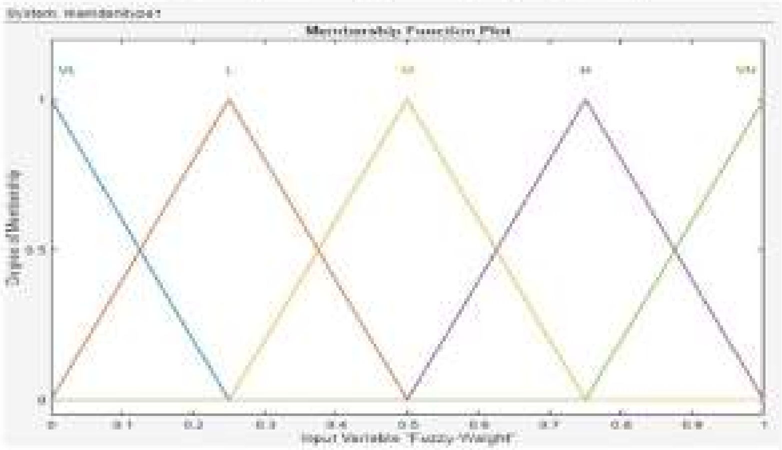


Figure 3
Triangular fuzzy membership function (fuzzy rate) as input

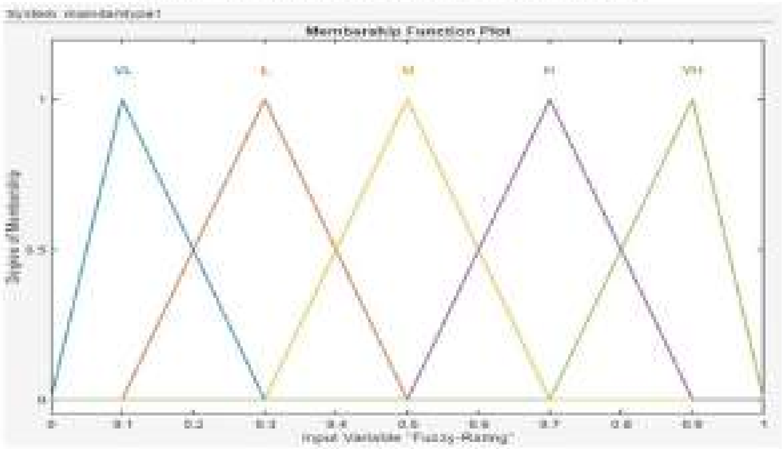


Figure 4
Triangular fuzzy membership function (overall fuzzy rating for novice user) as output

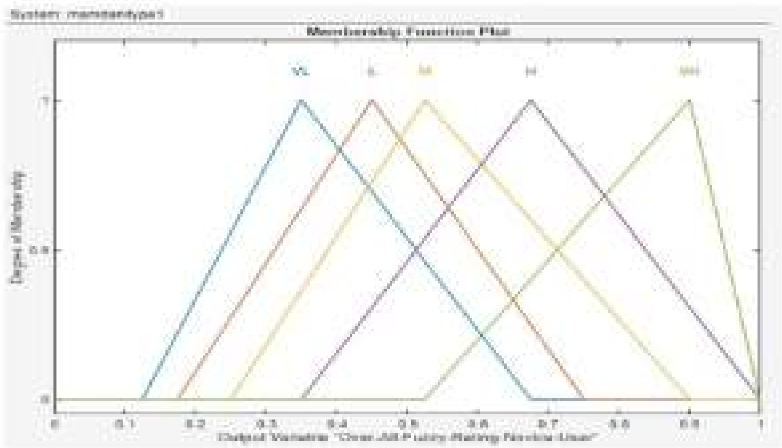
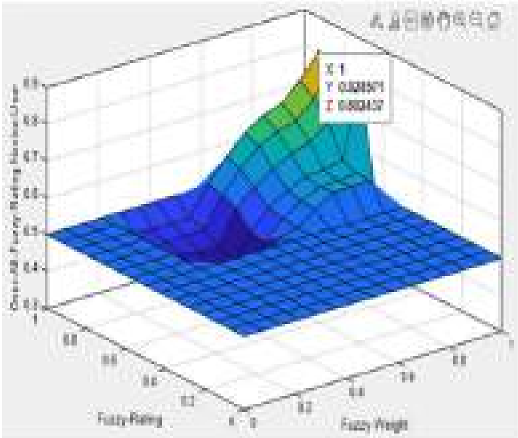


Figure 5
Control surface



7. Conclusion and Future Scope

This research study presents an innovative, effective, and novel technique for evaluating fuzzy weight for software quality criteria for mobile gaming applications based on the perspective of beginner users. This method assists stakeholders in mobile application organizations in achieving their aim of correct evaluation. The findings indicate appropriate verification, with the help of this validation; the technique that was chosen for this investigation was directed in the appropriate direction. This will also guarantee that the software industry evaluates the numerical value of the quality of mobile applications throughout the development process with more precision or accuracy and will thus take a few steps forward in the direction of an accurate assessment of the overall quality of mobile applications. Existing studies will benefit from this addition of a new dimension, especially those mobile applications that place a significant emphasis on accuracy or precision with their features. The present body of literature does not take into account the fact that crisp sum is one while assessing fuzzy weight; without a doubt, this will establish a new standard within the existing body of literature.

This research makes use of both the fuzzy set theory and the AHP to describe how new users' perspectives on the software quality elements in mobile games are inherently uncertain and imprecise. The use of FAHP is one of the most effective methods for accomplishing this goal. Users and the mobile game industry as a whole both stand to benefit from this development. The use of automation may allow for this mathematical model to be developed to a higher degree. The FAHP is a useful tool for tackling difficult decision-making issues that include a number of criteria and potential solutions. It is likely to predict that there will be an increase in the need for advanced decision-support technologies such as fuzzy AHP as decision-making challenges get more complicated. For better judgment, fuzzy AHP may be used with Artificial Intelligence and Machine Learning algorithms. It may help optimize machine learning processes by aiding in feature selection, model assessment, and adjusting machine learning parameters.

Recommendation

The use of quantitative data ensures that decisions are made on an objective basis. It is helpful in determining which areas need work, ranking the importance of additions or updates, and improving the efficiency of the development process. The software development company will be able to set clear and quantitative success measures with the assistance of this. This makes it possible to establish concrete objectives and monitor progress over the course of time. Quantitative data may also assist in the more efficient allocation of available resources. You may direct your efforts to the areas that deliver the best return on investment if you measure the effect of the different additions or modifications you implement.

Organizations may employ techniques like surveys, user analytics, user testing, and feedback-gathering tools in order to translate qualitative features into quantitative data. All of these approaches are responsible for assessing fuzzy ratings. The significance of fuzzy weight is contingent on the stakeholders and investors coming to a consensus. Therefore, the correctness of the computation is dependent on the value of the fuzzy weight. The use of fuzzy AHP helps to assure accuracy. This will give an appropriate foundation for the achievement of correct assessment. Therefore, it is recommended to all those enterprises, who produce mobile applications, that this framework approaches toward adequate computation. This will guarantee that app development organizations are able to give a comprehensive knowledge of their app's performance and user happiness via reliable quantitative data. As a result, it will be possible to enhance trust among investors and stakeholders.

It is possible for the researchers to use this method in order to determine various sets of weights according to the various perspectives held by stakeholders of the mobile gaming company. Subsequently, they will determine which prototype governs to have a more qualitative aspect by calculating the overall quality of various alternatives of the prototype of the mobile game application while mobile application development is taking place.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data available on request from the corresponding author upon reasonable request.

References

[1] Boehm, B. W., Brown, J. R., & Lipow, M. L. (1976). Quantitative evaluation of software quality. *Proceedings of the 2nd International Conference on Software Engineering*.

[2] Dromey, R. G. (1995). A model for software product quality. *IEEE Transactions on Software Engineering*, *21*(2), 146–162.

[3] ISO/IEC 9126-1. (2001). *Software engineering-product quality—Part 1: Quality model, Int'l organization for standardization*. Retrieved from: www.iso.org

[4] Pattnaik, S., Pattanayak, B., & Patnaik, S. (2019). Software quality prediction using fuzzy logic technique. International *Journal of Information Systems in the Service Sector, 11*, 51–71. https://doi.org/10.4018/IJISSS.2019040104

[5] Srivastava, P. R., Singh, A. P., & Vageesh, K. V. (2010). Assessment of software quality: A fuzzy multi-criteria approach. In *Evolution of Computation and Optimization Algorithms in Software Engineering: Applications and Techniques*. IGI Global USA, chapter – 11, 200–219.

[6] Dubey, S., & Sharma, D. (2015). Software quality appraisal using multi-criteria decision approach. *International Journal of Information Engineering and Electronic Business, 2*, 8–13.

[7] Reena, D., & Bhavesh, M. (2012). Measuring reliability of an aspect oriented software using fuzzy logic approach. *IJEAT, 1*, 233–237.

[8] Srivastava, P. R., & Kumar, K. (2009). An approach towards software quality assessment. *Communications in Computer and Information Systems Series (CCIS Springer Verlag), 31*(6), 345–346.

[9] Mishra, M., & Dadhich, R. (2023). How to quantify software quality factors for mobile applications?: Proposed criteria. *Emerging Trends in Expert Applications and Security: ICE-TEAS 2023. Lecture Notes in Networks and Systems, 681*. Singapore: Springer.

[10] Nitze, A., & Schmietendorf, A. (2015). A survey on mobile users' software quality perceptions and expectations. In *Software Testing, Verification and Validation Workshops, IEEE Eighth International Conference*, 1–2.

[11] Idri, A., Moumane, K., & Abran, A. (2013). On the use of software quality standard ISO/IEC 9126 in mobile environments. *IEEE 20th Asia-Pacific Software Engineering Conference*.

[12] Bevan, N. (1995). Usability is quality of use. In *Advances in Human Factors/Ergonomics, 20* (pp. 349–354). The Netherlands: Elsevier.

[13] Ghazizadeh, F. Z., & Vafadar, S. (2017). A quantitative evaluation of usability in mobile applications: An empirical study. In *Proceeding International Symposium on Computer Science and Software Engineering Conference*, 1–6.

[14] Cui, M., & Zhu, L. (2017). Usability evaluation methods of user interface based on mobile games using fuzzy methods. 124–131. https://doi.org/10.1007/978-3-319-53838-9_10

[15] Weichbroth, P. (2020). Usability of mobile applications: A systematic literature study. *IEEE Access, 8*, 55563–55577.

[16] Barnett, L., Harvey, C., & Gatzidis, C. (2018). Novice user experiences in mobile games: An evaluation of usability. *Entertainment Computing, 27*, 82–88.

[17] Saaty, T. L. (1980). *The analytic hierarchy process: Planning, priority setting, resource allocation*. USA: Mcgraw-Hill.

[18] Buckley, J. J. (1985). Fuzzy hierarchical analysis. *Fuzzy Sets and Systems, 17*(3), 233–247.

[19] Wu, H. Y., Tzeng, G. H., & Chen, Y. H. (2009). A fuzzy MCDM approach for evaluating banking performance based on balanced scorecard. *Expert Systems with Applications, 36*, 10135–10147.

[20] Sharafi, H., Soltanifar, M., &Lotfi, F. H. (2022).Selecting a green supplier utilizing the new fuzzy voting model and the fuzzy combinative distance-based assessment method. *EURO Journal on Decision Processes, 10*, 100010.

[21] Tavana, M., Soltanifar, M., Santos-Arteaga, F. J., & Sharafi, H. (2023). Analytic hierarchy process and data envelopment analysis: A match made in heaven. *Expert Systems with Applications, 223*, 119902.

[22] Hosseinzadeh Lotfi, F., Allahviranloo, T., Pedrycz, W., Shahriari, M., Sharafi, H., & Razipour GhalehJough, S. (2023). Weight determination methods in fuzzy environment. In *Fuzzy Decision Analysis: Multi Attribute Decision Making Approach. Studies in Computational Intelligence*, 1121. Springer.

[23] Nădăban, S., Dzitac, S., & Dzitac, I. (2016). Fuzzy TOPSIS: A general view. *Procedia Computer Science, 91*, 823–831, ISSN 1877-0509.

[24] Maryoly, O., Perez, M. A., & Rojas, T. (2003). Construction of a systemic quality model for evaluating software product. *Software Quality Journal, 11*(3), 219–242.

How to Cite: Mishra, M., & Dadhich, R. (2024). A Fuzzy AHP Approach to Evaluating the Fuzzy Weight of Mobile Game Application Software Quality Factors: Usability Expectations for Novice Users. *Archives of Advanced Engineering Science*. https://doi.org/10.47852/bonviewAAES42021934

Fuzzy Based Analysis of Software Quality Factor Understand-ability for Mobile Game Application

Manish Mishra **Reena Dadhich**

Department of Computer Science & Informatics
University of Kota
Kota, Rajasthan

ABSTRACT

Mobile gaming is becoming more popular, and end users demand fun and simple games. High-quality, user-friendly software is essential for keeping players pleased and returning back to mobile games. Mobile gaming stakeholders are constantly seeking for fresh applications to test and review while building new games. This project may benefit from measuring software quality-based qualitative attributes. Therefore, selecting the correct evaluation technique is crucial. Assessment uses fuzzy rate and fuzzy weight. The evaluation of a five-point fuzzy rating system needs simple survey or questionnaire inputs. The stakeholders who are accountable for the quality of the mobile app determine the fuzzy weight of metrics for a certain quality element. This is due to the fact that metrics are the fundamental building block for quantitative evaluation. When stakeholders analyze and contrast metrics for software quality factors and agree on their relevance, they may achieve the aim. This strategy will improve quality assessment overall. In this study, a novel Fuzzy based mathematical assessment method to access accurate fuzzy weight for the metrics of a particular quality factor will be provided in this paper. This knowledge may help the mobile gaming application industry build, evaluate and assign fuzzy weights of metrics, which will improve the translation of qualitative framework into quantitative ones. With the assistance of a fuzzy-based assessment approach and pair wise comparison, this study proposes a unique way for determining the relative ranking within the metrics for a certain quality factor. As a result, the appropriate fuzzy weight assigned to the metric. In this research, measurements of understand-ability quality factor of usability characteristic are taken into consideration, and a formulation developed to assign fuzzy weight to its metrics is presented with the assistance of a case study.

KEYWORDS: *Fuzzy logic, Metrics,Overall quality, Pair wise comparison, Quality factor etc.*

INTRODUCTION

The mobile gaming industry is one of the rapidly expanding fields that are seeing exponential growth over time. During the development process of any mobile application, the primary issue is the application's quality. End users are the assets that will become the promoters of the branding of any mobile application due to positive feedback. This is because end users are the ones that utilize the application. Usability is the characteristic that describes the quality according to the end users. As a result of the fact that usability is defined as a qualitative framework, it includes quality factors. It is not possible to take into consideration the qualitative framework for the purpose of assessment. The qualitative framework has to be transformed into a quantitative framework that can be measured as well. It was possible to accomplish this goal with the assistance of metric measurement that is associated with the quality factors. Using a five rating fuzzy scale system that ranges from VL (very low) to VH (very high), metrics is constructed in such a manner that they assess on a scale of five. Both the fuzzy weight and the fuzzy rate are considered to be fuzzy inputs. While fuzzy weight is determined by investors, quality managers, stakeholders, or what are generally referred to as decision makers in the mobile company, fuzzy rate is determined by the feedback that is provided in accordance with the prototype of the mobile application that is being developed. The purpose of this research was to offer a mathematical framework that is based on fuzzy logic and provides assistance to decision makers in determining the importance of fuzzy weight of metrics for a particular quality factor; this will be accomplished by the use of pair wise comparisons between the metrics of the particular quality factor. A fuzzy-based mathematical framework is analyzed and provided in this research. This framework is utilized to produce a correct relative ranking within the metrics of understandability quality factor. This ranking is then employed as an acceptable fuzzy weight for the metrics of understandability quality factor in accordance with the usability framework. The attainment of precise assessment and the achievement of accurate evaluation are both approaching one step closer for achieving an overall enhanced degree of accuracy.

LITERATURE REVIEW

The term "quality" is used to describe any product. Researchers have spent a lot of time researching quality models like Boehm's [1] and building simple models to describe quality. Maryol, Perez & Rojas [2] provided a quality model and software quality analysis approach. ISO, or the International Organization for Standardization, establishes quality standards globally. ISO 9126 [3] is one. These models are all- purpose; although they may be adapted to any software application by understanding its restrictions. Fuzzy logic is an effective technique to address evaluation uncertainty [4].

The next crucial question is how to quantify quantitative attributes. Quantification of quality factors analyses an approach to measure or justify software product quality. Researchers have shifted their attention significantly. Srivastavaet. alexamines fuzzy multi-criteria software quality measurement [5]. Software quality may be measured using multi-criteria decision-making [6]. Fuzzy logic may also assess aspect-oriented software stability [7] proposed by Dadhich & Mathur. Srivastava & Kumar [8] included project management, developer, and tester perspectives to the research. However, these studies still focus on developing a framework to ensure high-quality software and assessing it as desktop apps.

The next subject to be discussed is the framework for a mobile app that transforms the qualitative characteristics into a quantitative framework proposed by Mishra & Dadhich [9]. This research paper examined how mobile consumers see and expect software quality [10]. ISO 9126 now includes mobile situations [11].

FUZZY LOGIC

Fuzzy logic is a kind of logic that takes into account all of the values that fall between true and false, including the values that define its boundaries. The way that fuzzy logic operates is distinct from the way that crisp logic operates, which solely takes into account true and false values. This paper adopts triangular fuzzy membership function as shown in Figure 1. The mapping in between crisp value and fuzzy value illustrated in table1 and table 2.

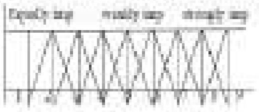


Fig. 1: Triangular Fuzzy Membership Function

Table 1. Relationship in Between Linguistic Variable & Fuzzy Value as per Figure 1

Compa-rison (crisp)	Linguistic variable	Fuzzy value
1	equally importance	(1,1,1)
2	In between equally importance & Weakly importance	(1,2,3)
3	Weakly importance	(2,3,4)
4	In between Weakly importance & Strongly importance	(3,4,5)
5	Strongly importance	(4,5,6)
6	In between Strongly importance & Very Strongly importance	(5,6,7)
7	Very Strongly importance	(6,7,8)
8	In between Very Strongly importance & Absolutely importance	(7,8,9)
9	Absolutely importance	(8,9,9)

Table 2. Relationship in Between Crisp Value & Inverse Fuzzy Value

Compa-rison (crisp)	Fuzzy value (Inverse)	Simplified fuzzy value
1/1	(1,1,1)-1	(1,1,1)
1/2	(1,2,3) -1	(1/3, 1/2, 1/1)
1/3	(2,3,4) -1	(1/4, 1/3, 1/2)
1/4	(3,4,5) -1	(1/5, 1/4, 1/3)
1/5	(4,5,6) -1	(1/6, 1/5, 1/4)
1/6	(5,6,7) -1	(1/7, 1/6, 1/5)
1/7	(6,7,8) -1	(1/8, 1/7, 1/6)
1/8	(7,8,9) -1	(1/9, 1/8, 1/7)
1/9	(8,9,9) -1	(1/9, 1/9, 1/8)

RESEARCH METHODOLOGY

Research Design

This paper presents a fuzzy-based assessment approach. The article considers five quality factors understand- ability, memorability, efficiency, challenge and user satisfaction as illustrated in table 3.

Table 3. Quality Factors to Access Usability

Characteristic	Quality Code	Quality Factors
Usability	QF1	Understand-ability
	QF2	Memorability
	QF3	Efficiency
	QF4	Challenge
	QF5	User satisfaction

This study employs a technique of conducting pair- wise comparisons among all metrics of understand-ability quality factor. The rationale for this approach is to determine the relative relevance of each metrics. Hence, this research constructs a matrix that compares metrics in pairs. This research focuses on the triangular fuzzy function.

Proposed Model

This paper adopts the algorithm discussed by (Nădăban & Dzitac, 2016, P-826) [12]. Following steps will explain the proposed model.

Step1. Design the usability framework for mobile game application.

Step2. Decide pair-wise priorities (crisp value) among metrics of particular quality factor.

Step3. Replace crisp value with corresponding fuzzy values as per linguistic variable.

Step4. Calculate fuzzy geometric mean (FGM) for all metrics of particular quality factor.

Step5. Calculate normalized fuzzy weight (NFW) from fuzzy geometric mean (FGM) for all metrics of particular quality factor.

Step6. Convert normalized fuzzy weight (NFW) in to corresponding crisp value and evaluate relative importance among all metrics of particular quality factor.

CASE STUDY

Suppose there is a mobile development organization 'ABC' whose stakeholders wants to evaluates relative ranking of metrics for understand-ability M1 to M5 and assign appropriate fuzzy weight as illustrated in table 4.

Table 4. Metrics M1 to M5 which defines and evaluate Quality Factor Understand-ability

Quality Factor	Metrics
Understand-ability	M1: icon of players are self-descriptive
	M2: reaction of environmental object as per user's action
	M3: time is feasible to complete each task
	M4: flow and logic of game easily understandable
	M5: skill set of players of mobile game are properly defined

Evaluate prioritization of metrics for QF1 Understand-ability

Following steps will explain the proposed model, which provides an explanation of the approach that may be used to determine the relative importance of all metrics by using pair-wise comparison methodology.

Step1: Design a pair-wise comparison framework matrix for each pair of metrics. Stakeholders & investors are mutually agreed upon following criteria imposed upon five metrics as illustrated in table5.

1. Metric M1 is 3 times more important than metric M3
2. Metric M1 has same importance as metric M5
3. Metric M2 is 3 times more important than metric M1
4. Metric M2 is 2 times more important than metric M3
5. Metric M2 is 3 times more important than metric M5
6. Metric M4 is 5 times more important than metric

M1,Metric M4 is 4 times more important than metric M2

7. Metric M4 is 4 times more important than metric M3
8. Metric M4 is 4 times more important than metric M5
9. Metric M4 is 2 times more important than metric M3

Table 5. Pair wise Comparison Matrix

	M1	M2	M3	M4	M5
M1	1	1/3	3	1/5	1
M2	3	1	2	1/4	3
M3	1/3	1/2	1	1/4	½
M4	5	4	4	1	4
M5	1	1/3	2	1/4	1

Step2. Replace fuzzy value by crisp value as per table 1 and table 2 and calculate fuzzy geometric mean (FGM) for each metric as illustrated in table6.

Computation of FGM for quality factor M1:

((1*1/4*2*1/6*1) 1/5, (1*1/3*3*1/5*1) 1/5, (1*1/2*4*1/4*1) 1/5) = (0.61,0.73,0.87)

Similarly calculated for other metrics M2 to M5.

Table 6. Computation of Fuzzy Geometric Mean (FGM) for each Metrics

	M1	M2	M3	M4	M5	FGM
M1	(1,1,1)	(1/4,1/3,1/2)	(2,3,4)	(1/6,1/5,1/4)	(1,1,1)	(0.61,0.73,0.87)
M2	(2,3,4)	(1,1,1)	(1,2,3)	(1/5,1/4,1/3)	(2,3,4)	(0.96,1.35,1.74)
M3	(1/4,1/3,1/2)	(1/3,1/2,1)	(1,1,1)	(1/5,1/4,1/3)	(1/3,1/2,1)	(0.35,0.46,0.69)
M4	(4,5,6)	(3,4,5)	(3,4,5)	(1,1,1)	(3,4,5)	(2.55,3.17,3.76)
M5	(1,1,1)	(1/4,1/3,1/2)	(1,2,3)	(1/5,1/4,1/3)	(1,1,1)	(0.55,0.69,0.87)

Step 3.Computation of normalized fuzzy weight for each metrics as illustrated in table7.

Computation of normalized fuzzy weight for metric M1:

(0.61, 0.73, 0.87) * (1/7.93, 1/6.4, 1/5.02) = (0.077, 0.114, 0.173)

Similarly calculated for other metrics.

Table 7 Computation of normalized fuzzy weight (NFW) for each Quality Factor

	FGM	NFW
M1	(0.61,0.73,0.87)	(0.077,0.114,0.173)
M2	(0.96,1.35,1.74)	(0.121,0.211,0.347)
M3	(0.35,0.46,0.69)	(0.044,0.72,0.137)
M4	(2.55,3.17,3.76)	(0.322,0.495,0.749)
M5	(0.55,0.69,0.87)	(0.069,0.108,0.173)

Step 4. Computation of crisp weight for each metrics, which also validate that crisp sum is 1.06 (almost 1). Step 4. Computation of crisp weight for each metrics, which also validate that crisp sum is 1.06 (almost 1). Crisp weight justifies overall relative importance in between the metrics for quality factor understand- ability as illustrated in table 8.

Step 5.Computation of relative importance with in metricsM1 to M5 as per crisp weight as illustrated in table 9 and table 10.

Step 6.Computation of fuzzy weight on the basis of relative distance in between metrics M1 to M5. One possible combination as illustrated table 11.

RESULT DISCUSSION

The ranking of the metrics M1 to M5 in relation to the crisp weight, which may be determined according on the order of importance outlined in Table 11. This confirmed findings and confirm a substantial correlation between the mathematical theoretical framework and the observed outcomes. The confirmed findings confirm a strong correlation between the theoretical framework and its actual implementation.

Table 8: Verification with Crisp Weight (Sum=1) and priorities of Metrics

	NFW	Crisp Weight	priority
M1	(0.077,0.114,0.173)	0.121	4th
M2	(0.121,0.211,0.347)	0.226	3rd
M3	(0.044,0.72,0.137)	0.300	2nd
M4	(0.322,0.495,0.749)	0.522	1st
M5	(0.069,0.108,0.173)	0.117	5th

Table 9: Relative importance with in metricsM1 to M5

	NFW	Crisp Weight	priority
M4	(0.322,0.495,0.749)	0.522	1st
M3	(0.044,0.72,0.137)	0.300	2nd
M2	(0.121,0.211,0.347)	0.226	3rd
M1	(0.077,0.114,0.173)	0.121	4th
M5	(0.069,0.108,0.173)	0.117	5th

Table 10. Allocation of fuzzy weight

Metric	Fuzzy weight
M4	VH
M3	H
M2	H
M1	M
M5	M

CONCLUSION & FUTURE SCOPE

This research study presents a technique with a special emphasis on the point of view of end users, which is both cutting-edge and efficient for evaluating the fuzzy weight of software metrics for a certain quality factor for mobile gaming applications. Through the use of this strategy, stakeholders in mobile application businesses are able to successfully achieve their objective of doing correct evaluation.

The study validates the methodology used for mobile app quality evaluation, ensuring precision in the software industry. It aims to improve the assessment of overall mobile application quality. The study will include a new component for high-priority mobile applications. It also highlights the potential of pairwise comparison with AI and ML algorithms for improved decision-making, feature selection, model assessment, and parameter modification, potentially improving machine understanding process optimization.

REFERENCES

1. Boehm B. W., Brown J. R. &Lipow M. L., Quantitative Evaluation of Software Quality, Proceedings of the 2nd International Conference on Software Engineering, San Francisco, CA, USA, 1976.
2. Maryoly O., Perez M.A. and Rojas T., Construction of a Systemic Quality Model for Evaluating Software Product, Software Quality Journal, Vol.11, No.3, 2003, 219-242.
3. ISO/IEC 9126-1:2001,Software Engineering-Product Quality—Part 1: Quality Model, Int'l Organization for Standardization, Available at www.iso.org.
4. Patnaik S, Pattanayak B &Patnaik S., Software Quality Prediction Using Fuzzy Logic Technique.International Journal of Information Systems in the Service Sector, 11, 2011, 51-71. 10.4018/IJISSS.2019040104.
5. Srivastava P. R., Singh, Vageesh K.V., Assessment of Software Quality: A Fuzzy Multi – Criteria Approach. Evolution of Computation and Optimization Algorithms in Software Engineering Applications and Techniques, IGI Global USA, chapter - 11, 2010, 200-219.
6. Dubey Sanjay, Sharma Disha (2015). Software quality Appraisal Using Multi-Criteria Decision Approach, Ijeeh, vol 2, 2015, 8-13.
7. DadhichRama, MathurShaveta, Measuring Reliability Of an Aspect Oriented Software Using Fuzzy Logic Approach, IJEAT, vol 1, 2012, 233-237.
8. Srivastava P. R. and Kumar K., An Approach towards Software Quality Assessment, Communications in Computer and Information Systems Series (CCIS Springer Verlag), Vol.31, No.6, 2009, 345-346.
9. Mishra M. &Dadhich R. How to Quantify Software Quality Factors for Mobile Applications?: Proposed Criteria. Emerging Trends in Expert Applications and Security. ICE-TEAS 2023. Lecture Notes in Networks and Systems, vol 681, Springer, Singapore, 2023.

How to Quantify Software Quality Factors for Mobile Applications?: Proposed Criteria

Manish Mishra and Reena Dadhich

Abstract The quality and reliability of any software product are the most important issues because the quality of any product directly depends upon branding. When one talks about software as a product then it is a big question of how to ensure every dimension of quality aspect. This issue becomes much bigger when moved from a desktop program to a mobile application, due to the inherent limitation of mobile due to mobility. Thus there is a requirement to help quality managers by customizing a suitable software quality model (e.g., ISO-9126, ISO-25010, etc.) standard to ensure software quality, in mobile environments. This paper proposed criteria to quantify the quality of a mobile application.

Keywords Quantification · Software quality · Mobile application · Fuzzy logic · Questionnaire · Survey etc

ICE-TEAS Springer 2023

2ND INTERNATIONAL CONFERENCE ON EMERGING TRENDS IN EXPERT APPLICATIONS & SECURITY

CERTIFICATE of PRESENTATION

This is to certify that

Mr Manish Mishra

has participated and presented a paper title

How To Quantify Software Quality Factors For Mobile Applications? : Proposed Criteria

2nd International Conference on Emerging Trends in Expert Applications & Security (ICE-TEAS 2023) to be held in **Jaipur Engineering College & Research Centre (JECRC), Jaipur** on **Feb 17-19, 2023.**

This conference held in hybrid mode.

Prof. Joao Manuel RS Tavares,	**Prof. Vinay Chandna**	**Prof. Vijay Singh Rathore**
PC Chair, ICE-TEAS 2023	General Chair-ICE-TEAS 2023	PC Chair & Convenor - ICE-TEAS 2023
Professir, University of Porto, Portugal	Principal, JECRC, Jaipur	Professor-CSE, JECRC, Jaipur

JECRC Foundation

Springer

A Pair-wise Comparison of Software Quality Factors for Prioritization of Fuzzy Weight using FAHP: Mobile Game Application Usability Expectations for Expert User

Manish Mishra[1], Reena Dadhich[2]

1 Department of Computer Science & Informatics, University of Kota, Kota, Rajasthan, India, kota.phd@gmail.com

2 Department of Computer Science & Informatics, University of Kota, Kota, Rajasthan, India, reena.dadhich@gmail.com

Abstract: A growing number of expert users are looking for entertaining and easy-to-understand experiences in the increasingly popular mobile gaming industry. Making ensuring these mobile games have high-quality, user-friendly software is crucial for keeping players happy and coming back for more. The subjective and sometimes vague expectations of expert users, however, are hard to evaluate. Stakeholders in the mobile gaming business are always looking for new apps to test and evaluate as they build new games. Measuring qualitative features that rely on software quality criteria can help in deploying this endeavor. Therefore, choosing the right assessment method is really critical. Two metrics, fuzzy rate and fuzzy weight, form the basis of the assessment. The assessment of a five-point fuzzy rating system requires simple inputs derived from surveys or questionnaires. Stakeholders tasked with guaranteeing suitable mobile application quality decide on fuzzy weight, which is shorthand for the relative importance of software quality factors along with priority. Stakeholders may reach this goal when they compare and contrast software quality elements and agree on their relative importance. The overall evaluation of quality will converge in the right direction with this method's help. In this paper, we provide a new usage of the Fuzzy Analytic Hierarchy Process (FAHP) to assess, for expert users in particular. The mobile game application sector may use this information to better formulate and assess fuzzy weights, which will strengthen the process of translating qualitative features into quantitative ones during mobile app development.

Keywords: software quality factors, mobile game, quantification, fuzzy logic, usability, expert users, FAHP.

JECRC Springer 2024 ICE-TEAS

3rd INTERNATIONAL CONFERENCE ON EMERGING TRENDS IN EXPERT APPLICATIONS & SECURITY

CERTIFICATE OF PRESENTATION

This is to certify that

Manish Mishra

has participated and presented a paper titled

A Pair-wise Comparison of Software Quality Factors for Prioritization of Fuzzy Weight using FAHP: Mobile Game Application Usability Expectations for Expert User

in 3rd International Conference on Emerging Trends in Expert Applications & Security (ICE-TEAS 2024)

held during 15-17 March 2024 at Jaipur Engineering College and Research Centre, Jaipur.

Prof. Vinay Kumar Chandna
General Chair-ICE-TEAS 2024
Principal, JECRC, Jaipur

Prof. Vijay Singh Rathore
PC Chair & Convenor - ICE-TEAS 2024

JAIPUR ENGINEERING COLLEGE AND RESEARCH CENTRE

Springer

Fuzzy Logic Based Quantification of Usability Expectation for Novice User in Mobile Games

Manish Mishra[1] **and Reena Dadhich**[2]

[1]*UOK, Kota*

[2] *Department of Computer Science and Informatics, UOK, Kota*

Usability is one of the important characteristics of any mobile game application. This characteristic is very important because users directly interact with a mobile application, so the opinion of any end user is important for the success of a mobile game application. Hence it is a big challenge to understand the end user's perspective on a mobile game, especially who interact with a mobile game first time i.e. novice users. Novice users should look for a game that is easy to learn, fun to play and has tutorials or instructions to help them get started. Mobile games should also be easy to use, have intuitive controls, and be visually appealing. Additionally, mobile games should be optimized for mobile devices to ensure that they run smoothly and quickly. Challenge and engagement are fundamental & core parts of the enjoyment of mobile games and are both dependent on usability. Thus it is required to explore sub-characteristics of usability, design its matrices and evaluate it numerically. Unlike most other mobile applications, the focus of this article proposes a framework for evaluating the usability of fuzzy logic-based systems for novice users. This paper explores sub-characteristics of usability along with the baseline model ISO 9126, designs the matrices, and evaluates based on novice users' responses. These evaluations are expressed as imprecise, vague linguistic expressions using a five-point Likert scale, very high to very low. The theory of fuzzy logic has been chosen as the most appropriate tool to deal with such kind of uncertainty.

Keywords: Usability, ISO 9126, Software Quality, Five point likert scale, Triangular Fuzzy Set, Fuzzy Multi Criteria Decision Approach etc.

AMITY UNIVERSITY

2[nd] International Conference on

Recent Trends in Engineering, Technology and Business Management (ICRTETBM 2023)

(Sustainability, Digitization and Business Operations)

Organized By Amity International Business School
Amity Centre for Interdisciplinary Research

23[rd] INTERNATIONAL BUSINESS HORIZON

INBUSH ERA WORLD SUMMIT 2023

22[nd] to 24[th] February 2023

CERTIFICATE OF PARTICIPATION

This is to certify that **Manish Mishra** from **Department of Computer Science and Informatics, UOK, Kota** has participated in International Conference on Recent Trends in Engineering, Technology and Business Management (ICRTETBM-2023) held from February 22[nd] to 24[th], 2023 at Amity University, Noida, India.

He/She also chaired a session / delivered a keynote / invited talk / presented a paper entitled **Fuzzy Logic Based Quantification of Usability Expectation for Novice User in Mobile Games**

Prof. (Dr.) Gurinder Singh
Group Vice Chancellor
General Chair, ICRTETBM 2023

Prof. (Dr.) P.K. Kapur
Director, ACIDR
Conference Chair, ICRTETBM 2023

Publishing Partner: Springer

Paper ID: 34

FUZZY LOGIC BASED USABILITY SUB-CHARACTERISTIC ANALYSIS FOR M-LEARNING APPLICATION UTILIZING GQM & ISO 9241-11

Manish Mishra[1], Reena Dadhich[2]

[1,2]Department of Computer Science & Informatics, University of Kota (UOK), Kota, Rajasthan

E-mail: [1]kota.phd@gmail.com, 2reena.dadhich@gmail.com

Abstract : The concept of usability is used to evaluate the user experience and the general impression of mobile apps by customers. Considering the significance of the mobile learning industry, the usability requirement for mobile apps employed in this field is very crucial. This is due to the fact that mobile learning is a crucial industry. This research will conduct a quantitative analysis of the usability sub-characteristics. This information will be used to provide feedback to stakeholders, developers, and testers, as well as to aid in the analysis and understanding of the different qualities that exist among the sub-characteristics. The goal of this paper was to provide a design framework that was created to assess the usability of a mobile learning application utilizing fuzzy logic. The assessment is conducted to ensure user satisfaction and enhance the overall quality of the M-Learning application.

Keywords: Usability, M-Learning, GQM Approach, ISO 9241-11, Survey, Questionnaire.

Taylor & Francis
Taylor & Francis Group

ICCTCCI - 2024

CUTTACK
ESTD -1962

INSTITUTE OF MANAGEMENT AND INFORMATION TECHNOLOGY, CUTTACK-753008 - INDIA

(A CONSTITUENT COLLEGE OF BPUT, GOVT. OF ODISHA)

This Certificate Is Proudly Presented To

Dr./Mr./Ms. Manish Mishra

from University of Kota (UOK) Kota, Rajasthan **presented the paper**

Fuzzy Logic Based Usability Sub-Characteristic Analysis for M-Learning Application Utilizing GQM & ISO 9241-11

in International Conference on

"Cutting Edge Technology in Computing, Communication and Intelligence"

during 23rd - 24th March 2024.

Dr. Suvendra Kumar Jayasingh
CONVENOR

Prof. Srinivas Sethi
PROGRAM CHAIR

APPENDIX – 1

SURVEY QUESTIONNAIRE FOR DEVELOPER

A questionnaire related to access the quality of M-Commerce application is prepared for the purpose of survey. This questionnaire or metrics contains questions primarily related to assess each software quality factors (SQFs) as per the developer. This questionnaire based on five scale fuzzy rating, VH to VL and ISO/IEC-9126 standard.

M-Commerce Application: Developer's view

This questionnaire is designed to assess an M-Commerce application as per the developer based on ISO/IEC-9126 and fuzzy-logic

1. Name of Developer

____

2. Name of Mobile App Development Company

____

3. *Mark only one oval per row.*

	VH	H	M	L	VL
MD1:Find Probability for no. of operations implemented	◯	◯	◯	◯	◯

4. *Mark only one oval per row.*

	VH	H	M	L	VL
MD2:Find Probability as per formula (each sensor)	◯	◯	◯	◯	◯

5.

Mark only one oval per row.

	VH	H	M	L	VL
MD3:Find Probability as per formula(multiple sensors)	⬭	⬭	⬭	⬭	⬭

6.

Mark only one oval per row.

	VH	H	M	L	VL
MD4:App behaves well for all H/W & OS.	⬭	⬭	⬭	⬭	⬭

7.

Mark only one oval per row.

	VH	H	M	L	VL
MD5:Behavior of each API functions	⬭	⬭	⬭	⬭	⬭

8.

Mark only one oval per row.

	VH	H	M	L	VL
MD6:Behavior of each sensor	⬭	⬭	⬭	⬭	⬭

9.

Mark only one oval per row.

	VH	H	M	L	VL
MD7:Adherence to functionality standard	⬭	⬭	⬭	⬭	⬭

10.

Mark only one oval per row.

	VH	H	M	L	VL
MD8:Degree of access controllability	⬭	⬭	⬭	⬭	⬭

11.

Mark only one oval per row.

	VH	H	M	L	VL
MD9:Mode of payment gateway	⬭	⬭	⬭	⬭	⬭

12.

Mark only one oval per row.

	VH	H	M	L	VL
MD10:Display as per device screen	⬭	⬭	⬭	⬭	⬭

13.

Mark only one oval per row.

	VH	H	M	L	VL
MD11:Font size and style as per UI provided by client	◯	◯	◯	◯	◯

14.

Mark only one oval per row.

	VH	H	M	L	VL
MD12:About landscape and portrait	◯	◯	◯	◯	◯

15.

Mark only one oval per row.

	VH	H	M	L	VL
MD13:Fill a complete form as per size of screen	◯	◯	◯	◯	◯

16.

Mark only one oval per row.

	VH	H	M	L	VL
MD14:About response time	◯	◯	◯	◯	◯

17.

Mark only one oval per row.

	VH	H	M	L	VL
MD15:Throughput Rate	⬭	⬭	⬭	⬭	⬭

18.

Mark only one oval per row.

	VH	H	M	L	VL
MD16:Memory and processor as per app	⬭	⬭	⬭	⬭	⬭

19.

Mark only one oval per row.

	VH	H	M	L	VL
MD17:Synchronization of H\W selection	⬭	⬭	⬭	⬭	⬭

20.

Mark only one oval per row.

	VH	H	M	L	VL
MD18:Tradeoff in between H/W selection and power dissipation	⬭	⬭	⬭	⬭	⬭

21.

Mark only one oval per row.

	VH	H	M	L	VL
MD19:Duration of execution of app and power dissipation	⬭	⬭	⬭	⬭	⬭

22.

Mark only one oval per row.

	VH	H	M	L	VL
MD20:N\W selection and execution of app	⬭	⬭	⬭	⬭	⬭

23.

Mark only one oval per row.

	VH	H	M	L	VL
MD21:Adherence to efficiency standard	⬭	⬭	⬭	⬭	⬭

24.

Mark only one oval per row.

	VH	H	M	L	VL
MD22:Support for multiple user	⬭	⬭	⬭	⬭	⬭

25.

Mark only one oval per row.

	VH	H	M	L	VL
MD23:What if users randomly increased?	◯	◯	◯	◯	◯

26.

Mark only one oval per row.

	VH	H	M	L	VL
MD24:Mobility Test	◯	◯	◯	◯	◯

27.

Mark only one oval per row.

	VH	H	M	L	VL
MD25:0ptimal threading model to avoid suspension or shut down of thread	◯	◯	◯	◯	◯

28.

Mark only one oval per row.

	VH	H	M	L	VL
MD26:Algo to implement priority among threads	◯	◯	◯	◯	◯

29.

Mark only one oval per row.

	VH	H	M	L	VL
MD27:App with data to display initially, even if low or no connectivity	⬭	⬭	⬭	⬭	⬭

30.

Mark only one oval per row.

	VH	H	M	L	VL
MD28:Use of Caching scheme and local storage of data	⬭	⬭	⬭	⬭	⬭

31.

Mark only one oval per row.

	VH	H	M	L	VL
MD29:Special care about data dependent application	⬭	⬭	⬭	⬭	⬭

32.

Mark only one oval per row.

	VH	H	M	L	VL
MD30:Opportunity to adapt app in android as well as IOS	◯	◯	◯	◯	◯

33.

Mark only one oval per row.

	VH	H	M	L	VL
MD31:Optimal combination of H/W for all devices and for all platforms	◯	◯	◯	◯	◯

34.

Mark only one oval per row.

	VH	H	M	L	VL
MD32:Relative ease of installing software for a given platform	◯	◯	◯	◯	◯

35.

Mark only one oval per row.

	VH	H	M	L	VL
MD33:Frequency of deadlocks	◯	◯	◯	◯	◯

36.

Mark only one oval per row.

	VH	H	M	L	VL
MD34:Adherence to Portability standard	⬭	⬭	⬭	⬭	⬭

37.

Mark only one oval per row.

	VH	H	M	L	VL
MD35:KLOC (size oriented metric)	⬭	⬭	⬭	⬭	⬭

38.

Mark only one oval per row.

	VH	H	M	L	VL
MD36:Skills	⬭	⬭	⬭	⬭	⬭

39.

Mark only one oval per row.

	VH	H	M	L	VL
MD37:Experience	⬭	⬭	⬭	⬭	⬭

40.

Mark only one oval per row.

	VH	H	M	L	VL
MD38:Documentation	⬭	⬭	⬭	⬭	⬭

41.

Mark only one oval per row.

	VH	H	M	L	VL
MD39:Applicability of CMM level	⬭	⬭	⬭	⬭	⬭

42.

Mark only one oval per row.

	VH	H	M	L	VL
MD40:No of global variables	⬭	⬭	⬭	⬭	⬭

43.

Mark only one oval per row.

	VH	H	M	L	VL
MD41:Support error handling or exception handling	⬭	⬭	⬭	⬭	⬭

44.

Mark only one oval per row.

	VH	H	M	L	VL
MD42:Adherence to Maintainability standard	⬭	⬭	⬭	⬭	⬭

45.

Mark only one oval per row.

	VH	H	M	L	VL
MD43:Specific level of performance	⬭	⬭	⬭	⬭	⬭

46.

Mark only one oval per row.

	VH	H	M	L	VL
MD44:Performance after updation	⬭	⬭	⬭	⬭	⬭

47.

Mark only one oval per row.

	VH	H	M	L	VL
MD45:Degree of recoverability	⬭	⬭	⬭	⬭	⬭

48.

Mark only one oval per row.

	VH	H	M	L	VL
MD46:Adherence to Reliability standard	⬭	⬭	⬭	⬭	⬭

APPENDIX – 2

SURVEY QUESTIONNAIRE FOR TESTER

A questionnaire related to access the quality of M-Commerce application is prepared for the purpose of survey. This questionnaire or metrics contains questions primarily related to assess each software quality factors (SQFs) as per the tester. This questionnaire based on five scale fuzzy rating, VH to VL and ISO/IEC-9126 standard.

M-Commerce Application: Tester's view

This questionnaire is designed to assess an
M-Commerce application as per the tester based on ISO/IEC-9126 and fuzzy-logic

1. Name of Tester

____

2. Name of Mobile App Development Company

____

3. *Mark only one oval per row.*

	VH	H	M	L	VL
MT1:Implementation of all functions	◯	◯	◯	◯	◯

4. *Mark only one oval per row.*

	VH	H	M	L	VL
MT2:Behavior of sensors	◯	◯	◯	◯	◯

5. *Mark only one oval per row.*

	VH	H	M	L	VL
MT3:Precision of computation	◯	◯	◯	◯	◯

6.

Mark only one oval per row.

	VH	H	M	L	VL
MT4:Easily installed first time for android and ios	⬭	⬭	⬭	⬭	⬭

7.

Mark only one oval per row.

	VH	H	M	L	VL
MT5:Uninstalled, then again installed easily for android and ios	⬭	⬭	⬭	⬭	⬭

8.

Mark only one oval per row.

	VH	H	M	L	VL
MT6:Behavior of multiple sensors	⬭	⬭	⬭	⬭	⬭

9.

Mark only one oval per row.

	VH	H	M	L	VL
MT7:Adherence to Functionality standard	⬭	⬭	⬭	⬭	⬭

10.

Mark only one oval per row.

	VH	H	M	L	VL
MT8:Degree of access controllability	◯	◯	◯	◯	◯

11.

Mark only one oval per row.

	VH	H	M	L	VL
MT9:Overall content of application including buttons displayed as per device screen	◯	◯	◯	◯	◯

12.

Mark only one oval per row.

	VH	H	M	L	VL
MT10:Font size and style	◯	◯	◯	◯	◯

13.

Mark only one oval per row.

	VH	H	M	L	VL
MT11:Support both landscape and portrait mode	⬭	⬭	⬭	⬭	⬭

14.

Mark only one oval per row.

	VH	H	M	L	VL
MT12:Optimized to fill a complete form as per size of screen	⬭	⬭	⬭	⬭	⬭

15.

Mark only one oval per row.

	VH	H	M	L	VL
MT13:Response time as per devices and network	⬭	⬭	⬭	⬭	⬭

16.

Mark only one oval per row.

	VH	H	M	L	VL
MT14:App behavior as per battery	⬭	⬭	⬭	⬭	⬭

17.

Mark only one oval per row.

	VH	H	M	L	VL
MT15:Level of battery as per time duration	⬭	⬭	⬭	⬭	⬭

18.

Mark only one oval per row.

	VH	H	M	L	VL
MT16:Adherence to efficiency standard	⬭	⬭	⬭	⬭	⬭

19.

Mark only one oval per row.

	VH	H	M	L	VL
MT17:Support multiple users	⬭	⬭	⬭	⬭	⬭

20.

Mark only one oval per row.

	VH	H	M	L	VL
MT18:Load balancing if users randomly increases	⬭	⬭	⬭	⬭	⬭

21.

Mark only one oval per row.

	VH	H	M	L	VL
MT19:Responce of Mobility test	⬭	⬭	⬭	⬭	⬭

22.

Mark only one oval per row.

	VH	H	M	L	VL
MT20:Loading time for app	⬭	⬭	⬭	⬭	⬭

23.

Mark only one oval per row.

	VH	H	M	L	VL
MT21:Functioning when switch to other app	⬭	⬭	⬭	⬭	⬭

24.

Mark only one oval per row.

	VH	H	M	L	VL
MT22:Phone calls do not create obstacle in functioning of app	⬭	⬭	⬭	⬭	⬭

25.

Mark only one oval per row.

	VH	H	M	L	VL
MT23:Response time as per devices and N\W	⬭	⬭	⬭	⬭	⬭

26.

Mark only one oval per row.

	VH	H	M	L	VL
MT24:Data available in first time	⬭	⬭	⬭	⬭	⬭

27.

Mark only one oval per row.

	VH	H	M	L	VL
MT25:Updation easily in app	⬭	⬭	⬭	⬭	⬭

28.

Mark only one oval per row.

	VH	H	M	L	VL
MT26:Response time for each platform	⬭	⬭	⬭	⬭	⬭

29.

Mark only one oval per row.

	VH	H	M	L	VL
MT27:Installation form source in android & IOS	⬭	⬭	⬭	⬭	⬭

30.

Mark only one oval per row.

	VH	H	M	L	VL
MT28: Is app hangs for each platform?	⬭	⬭	⬭	⬭	⬭

31.

Mark only one oval per row.

	VH	H	M	L	VL
MT29:Adherence to portability standard	⬭	⬭	⬭	⬭	⬭

32.

Mark only one oval per row.

	VH	H	M	L	VL
MT30: Overall Responce of mobile App	⬭	⬭	⬭	⬭	⬭

33.

Mark only one oval per row.

	VH	H	M	L	VL
MT31:Degree of stability	⬭	⬭	⬭	⬭	⬭

34.

Mark only one oval per row.

	VH	H	M	L	VL
MT32:Adherence to maintainability standard	⬭	⬭	⬭	⬭	⬭

35.

Mark only one oval per row.

	VH	H	M	L	VL
MT33:Breakdown avoidance	⬭	⬭	⬭	⬭	⬭

36.

Mark only one oval per row.

	VH	H	M	L	VL
MT34:Degree of recoverability	⬭	⬭	⬭	⬭	⬭

37.

Mark only one oval per row.

	VH	H	M	L	VL
MT35:Adherence to reliability standard	◯	◯	◯	◯	◯

38.

Mark only one oval per row.

	VH	H	M	L	VL
MT36:Easy to modify as per updated versions	◯	◯	◯	◯	◯

39.

Mark only one oval per row.

	VH	H	M	L	VL
MT37:Easy to modify as per removal of errors	◯	◯	◯	◯	◯

40.

Mark only one oval per row.

	VH	H	M	L	VL
MT38:Test cases are optimal	◯	◯	◯	◯	◯

41.

Mark only one oval per row.

	VH	H	M	L	VL
MT39:Cyclomatic complexity	◯	◯	◯	◯	◯

42.

Mark only one oval per row.

	VH	H	M	L	VL
MT40: (functional, performance, api , usability etc.) have to be performed	◯	◯	◯	◯	◯

43.

Mark only one oval per row.

	VH	H	M	L	VL
MT41:Manual and automated testing both performed	◯	◯	◯	◯	◯

44.

Mark only one oval per row.

	VH	H	M	L	VL
MT42:Failure density against test cases	◯	◯	◯	◯	◯

45.

Mark only one oval per row.

	VH	H	M	L	VL
MT43:Failure resolution	◯	◯	◯	◯	◯

APPENDIX – 3

SURVEY QUESTIONNAIRE FOR β-USERS

A questionnaire related to access the quality of M-Commerce application is prepared for the purpose of survey. This questionnaire or metrics contains questions primarily related to assess each software quality factors (SQFs) as per the β-users. This questionnaire based on five scale fuzzy rating, VH to VL and ISO/IEC-9241-11 standard. This questionnaire based upon GQM approach.

Usability Evaluation for M-Commerce Application: β-User's View

This questionnaire based upon usability
evaluation (GQM Approach based on ISO/IEC-9241-11 and fuzzy-logic)

1. Name of Respondent

____

2. Age of Respondent

____

3. Gender of Respondent

Mark only one oval.

◯ Male
◯ Female

4.

Mark only one oval per row.

	VH	H	M	L	VL
MβU1:Consistency in text, font and colors	◯	◯	◯	◯	◯

5.

Mark only one oval per row.

	VH	H	M	L	VL
MβU2:Easy to navigation in mobile app	⬭	⬭	⬭	⬭	⬭

6.

Mark only one oval per row.

	VH	H	M	L	VL
MβU3:Support night vision	⬭	⬭	⬭	⬭	⬭

7.

Mark only one oval per row.

	VH	H	M	L	VL
MβU4:Finding information about product are easier	⬭	⬭	⬭	⬭	⬭

8.

Mark only one oval per row.

	VH	H	M	L	VL
MβU5:Speech to text converted accurately	⬭	⬭	⬭	⬭	⬭

9.

Mark only one oval per row.

	VH	H	M	L	VL
MβU6:Description of each product is accurate	◯	◯	◯	◯	◯

10.

Mark only one oval per row.

	VH	H	M	L	VL
MβU7:Product photograph displayed adequately	◯	◯	◯	◯	◯

11.

Mark only one oval per row.

	VH	H	M	L	VL
MβU8:Prices of products are adequately shown	◯	◯	◯	◯	◯

12.

Mark only one oval per row.

	VH	H	M	L	VL
MβU9:Status (available, out of stock) of each product is adequately shown	◯	◯	◯	◯	◯

13.

Mark only one oval per row.

	VH	H	M	L	VL
MβU10 Easy to register	⬭	⬭	⬭	⬭	⬭

14.

Mark only one oval per row.

	VH	H	M	L	VL
MβU11:Easy to change customer information	⬭	⬭	⬭	⬭	⬭

15.

Mark only one oval per row.

	VH	H	M	L	VL
MβU12:Easy to order product	⬭	⬭	⬭	⬭	⬭

16.

Mark only one oval per row.

	VH	H	M	L	VL
MβU13:Shopping cart's information is accurate	⬭	⬭	⬭	⬭	⬭

17.

Mark only one oval per row.

	VH	H	M	L	VL
MβU14:Adequate information about how to order	⬭	⬭	⬭	⬭	⬭

18.

Mark only one oval per row.

	VH	H	M	L	VL
MβU15:Adequate information about payment options	⬭	⬭	⬭	⬭	⬭

19.

Mark only one oval per row.

	VH	H	M	L	VL
MβU16:Adequate information about how to cancel the product	⬭	⬭	⬭	⬭	⬭

20.

Mark only one oval per row.

	VH	H	M	L	VL
MβU17:Adequate information about return & refund policy	⬭	⬭	⬭	⬭	⬭

21.

Mark only one oval per row.

	VH	H	M	L	VL	Column 6
MβU18:Adequate information about order detail	⬭	⬭	⬭	⬭	⬭	⬭

22.

Mark only one oval per row.

	VH	H	M	L	VL
MβU19 Adequate information about delivery time	⬭	⬭	⬭	⬭	⬭

23.

Mark only one oval per row.

	VH	H	M	L	VL
MβU20:Adequate information about delivery cost	⬭	⬭	⬭	⬭	⬭

24.

Mark only one oval per row.

	VH	H	M	L	VL
MβU21:Adequate information about delivery area	⬭	⬭	⬭	⬭	⬭

25\.

Mark only one oval per row.

	VH	H	M	L	VL
MβU22:Delivery to other address	◯	◯	◯	◯	◯

26\.

Mark only one oval per row.

	VH	H	M	L	VL
MβU23:Online order tracking available	◯	◯	◯	◯	◯

27\.

Mark only one oval per row.

	VH	H	M	L	VL
MβU24:Search particular product	◯	◯	◯	◯	◯

28\.

Mark only one oval per row.

	VH	H	M	L	VL
MβU25:Time to complete a task	◯	◯	◯	◯	◯

29.

Mark only one oval per row.

	VH	H	M	L	VL
MβU26:Easy to learn interface	⬭	⬭	⬭	⬭	⬭

30.

Mark only one oval per row.

	VH	H	M	L	VL
MβU27:Adequate content management	⬭	⬭	⬭	⬭	⬭

31.

Mark only one oval per row.

	VH	H	M	L	VL
MβU28:Adequate help (demo version)	⬭	⬭	⬭	⬭	⬭

32.

Mark only one oval per row.

	VH	H	M	L	VL
MβU29:Adequate help (text version)	⬭	⬭	⬭	⬭	⬭

33.

Mark only one oval per row.

	VH	H	M	L	VL
MβU30:Mobile app respond properly as per action	⬭	⬭	⬭	⬭	⬭

34.

Mark only one oval per row.

	VH	H	M	L	VL
MβU31:Every component of interface respond accurately	⬭	⬭	⬭	⬭	⬭

35.

Mark only one oval per row.

	VH	H	M	L	VL
MβU32:Probability to search successfully task completion in 1st attempt	⬭	⬭	⬭	⬭	⬭

36.

Mark only one oval per row.

	VH	H	M	L	VL
MβU33:Probability to completion of task within given time	⬭	⬭	⬭	⬭	⬭

37.

Mark only one oval per row.

	VH	H	M	L	VL
MβU34:Adequate information about privacy policy	⬭	⬭	⬭	⬭	⬭

38.

Mark only one oval per row.

	VH	H	M	L	VL
MβU35:Secure socket layer used by mobile app	⬭	⬭	⬭	⬭	⬭

39.

Mark only one oval per row.

	VH	H	M	L	VL
MβU36:Well recognized secure payment methods	⬭	⬭	⬭	⬭	⬭

40.

Mark only one oval per row.

	VH	H	M	L	VL
MβU37:Different mode for verification such as OTP based	⬭	⬭	⬭	⬭	⬭

41. *Mark only one oval per row.*

	VH	H	M	L	VL
MβU38:Overall features	◯	◯	◯	◯	◯

42. *Mark only one oval per row.*

	VH	H	M	L	VL
MβU39:Overall learning process	◯	◯	◯	◯	◯

43. *Mark only one oval per row.*

	VH	H	M	L	VL
MβU40:Overall accuracy	◯	◯	◯	◯	◯

44. *Mark only one oval per row.*

	VH	H	M	L	VL
MβU41:Overall security	◯	◯	◯	◯	◯

45. *Mark only one oval per row.*

	VH	H	M	L	VL
MβU42:Overall experience to use this mobile app	◯	◯	◯	◯	◯