A Thesis

on

Fuzzy Multi Criteria Decision Analysis for Security Durability Assessment

by

Rajeev Kumar

Department of Information Technology

Submitted in fulfillment of the requirement of degree of

Doctor of Philosophy

to the

Babasaheb Bhimrao Ambedkar University
(A Central University)
Lucknow, Uttar Pradesh, India

December – 2018

DECLARATION

I, Rajeev Kumar, solemnly declare that this thesis of research on "**Fuzzy Multi Criteria Decision Analysis for Security Durability Assessment**" is my original work. The study has been conducted under the guidance of Prof. Raees Ahmad Khan and Dr. Suhel Ahmad Khan, at Department of Information Technology, Babasaheb Bhimrao Ambedkar University (A Central University), Lucknow. It is further declared that to the best of my knowledge and belief it has not been submitted earlier for the award of any degree. I also undertake that the thesis is essentially free from all kinds of plagiarism.

Dated:

(**Rajeev Kumar**)

Researcher
Department of Information Technology
Babasaheb Bhimrao Ambedkar University
(A Central University)
Lucknow, Uttar Pradesh, India

CERTIFICATE

This is to certify that the thesis entitled ‘‘**Fuzzy Multi Criteria Decision Analysis for Security Durability Assessment**’’ submitted by **Mr. Rajeev Kumar** is an original research work and has not been previously submitted in part or full for the award of any other degree or diploma to this or any other University.

This thesis submitted to Babasaheb Bhimrao Ambedkar University Lucknow satisfies all the requirements as stipulated in the *Doctor of Philosophy (Ph.D.)* regulations-1999 as amended in 2013 and it is fit for submission and evaluation for the award of the degree of Doctor of Philosophy of the University.

Co-Supervisor **Supervisor**

Dated: **Head of the Department**

ACKNOWLEDGEMENTS

The tedious journey of life comes as blessing with the objective of learning by following the fruitful path bestowed with many enigmas and also with many results. All should be rested with the Almighty as I am not alone and will not remain alone because it is He who made this path full of hopes and learnings and turns anordinary man into a man of reason and rationality. This thesis is very insignificant contribution in the vast ocean of knowledge whose care taker is Almighty.

I am indeed indebted to my distinguished guide and supervisor Prof. Raees Ahmad Khan for all his help during the course of the study. Certainly, I am short of words to convey my real feelings for his invaluable help and concern together with 'scholarly' insightful and 'critical' guidance. I do not hesitate to state that without his help it would not have been possible for me to complete the research work. I am very grateful to my co-supervisor Dr. Suhel Ahmad Khan for his guidance, support and consultations during the course of the study.

I am also thankful to Dr. Alka, for her continuous encouragement, guidance, moral support, and consultations during the course of the study. I am thankful to SAQ Infosys for granting me the permission and needed assistance for data collection and experiments. I am also thankful to all my friends and colleagues for providing encouragement and support especially Surabhi Dwivedi, Jasleen Kaur, Richa Verma, Mohd Waris Khan, Mohd Faizan and Kavita Sahu. I express my sincere thanks to all faculty members and office staff of the department for their time-to-time continuous encouragement and support. I express my sincere thanks to all the experts from India and abroad for honoring me with their valuable observations during the Expert Opinions.

Last but not least, I cannot afford to forget to express my indebtedness to my parents (Shri Anil Kumar Srivastava and Smt. Indu Devi), brothers and sisters. I have neglected some of the duties towards them and appreciate their tolerance even in the worst circumstances. On the contrary, they have been a source of inspiration and consolation for me.

Rajeev Kumar

ABSTRACT

Security of user's information is at risk, as the increasing use of software makes it important to use software in every field. Nowadays, it is easy to build and use the software but to maintain its security is not an easy task because organizations are facing numerous issues related to security services of software. This introduces an urgent need to address security issues as security failure may lead to disastrous effects on human lives. Complex operations, rising cost, resource constraint, and a future of strategic uncertainty demand that software must deliver higher security with reducing cost. This will help in building software that will actually be able to defend itself from attacks despite being dependent upon any application security software (say, *antivirus*) for its protection against threats. The basic cause of the maximum of the security breaches is the presence of loopholes in the end product. The early detection and correction of these ambiguities may help reduce the occurrence of such attacks. In order to reduce the occurrence of security violations, it becomes indispensable to address the security issues during software development life cycle. Software developers are trying their best to achieve higher security of software. But, security of software is still not at its best. In addition, organizations are demanding optimal maintenance of security during working life of software services.

To fulfil the organization's demand, practitioners are always in search of better ways to manage security services for long duration. There is no straight forward solution available for problems of improving life span of security. Further, practitioners are trying to achieve durable software but unfortunately, they are ignoring the concept of security durability. Without a deep research of security durability, there is no way to get durable performance of software. If durable software is not secure then user will loss his/her trust on software. That is why, security of software is as much important as software durability. Hence, this makes the efforts of developing durable software worthless. After thoroughly reviewing literature, it is found that there is no work available in the area related to security durability assessment. With the critical examination of literature survey and best practices, Security durability is defined *as the duration during which the software performs securely*. Without paying attention on security durability, the software may start failing after deployment. Further, ignoring security durability may badly

affect service life of software. In addition, less durable security of software is likely to fail in the market.

It can be analysed that assessment of security durability is a significant step to improve the security, and without its consideration, potential of CIA (Confidentiality, Integrity and Availability) cannot be enhanced for a specific time period. Assessment of security durability is not possible without understanding the relation between security and durability. Further, durability of security services depends on budget and maintenance time. If it is possible to assess the durability of security services, the cost and time of maintenance would have been optimized. Security services of software must be longer with optimal maintenance as insecure services of software will gravitate to the insecure alternative. Security durability assessment is an important step towards improving durability of security as well as software services. Further, security durability measures may include one or more factors or attributes in it. To evaluate the security durability of software, there is need to assess the attributes which are related to security durability, directly or indirectly. In this row, the current research is done with three components in it, which are; development of framework, implementing the framework and assessment and third component is validation,done empirically as well as theoretically.

The first component referred to the development of the framework to identify security and durability attributes and its sub-attributes that affects security life span directly or indirectly. Correlate these attributes in order to assess the security durability of software. A framework for security durability assessment has been accomplished through the literature survey, gathering opinion from the practitioners, needed development, validations and revisions. The conceptualization phase is a brainstorming activity to precisely understand the problem and to gather related facts. Planning provides the roadmap to the design based on information from conceptualization phase. Designing is the most important and critical step towards the development of security models. Validation provides the supporting evidence as to whether a measure really captures the internal attributes that it purports to measure. Review and Revision phase facilitates the activity of ‘look back and change’, if required with a free-to-entre option to any of earlier phase.

The second component of the study is to implement the proposed framework for security durability assessment. In order to provide the significant and improved measurement of security durability, it is required to relate the durability attributes and desirable security attributes. Researcher establishes a correlation between durability and security attributes using Multiple Criteria Decision Analysis (MCDA) or Multiple Criteria Decision Making (MCDM) technique. Both hybrid and classical methods are used for assessment in this study, because, decision making process is a complicated phenomenon. Entrance Examination Software for Babasaheb Bhimrao Ambedkar University (A Central University), Lucknow, India (BBAU Software) is examined for assessing security services throughout the research work. Security services of BBAU Software are very crucial and important due to sensitive information of online entrance exam. The results of security durability assessment may help developers to improve longevity of secure software after development. Security durability consideration might help in reducing the maintenance effort incurred on security life span of software services.

The third component of the study is to confirm that how developed security durability assessment model is helpful for improvement of security life span of software services. Suggestions/rules/procedures are essential activities during development for improving the security service life span. It promotes the reengineering measures for improving working life of security as well as software services. The researcher made an effort in this regard to develop suggestions for longer security services. The given suggestions are helpful to manage security for longer life span. The proposed model calculates the security durability and revised version of BBAU Software is being influenced through suggestions. Sensitivity analysis analyzed to show the variations in results due to changing in values. Further, the validated results of statistical analysis with case study that reflect the usefulness and acceptability of developed model and suggestions is tested with hypothesis testing. The null hypothesis is strongly rejected on alpha level of significance for two tailed test. Hence, alternate hypothesis at a very good level of significance are accepted for improvements of security service life span.

This research work is done in the area of security life span and security is one of the biggest concerns in today's era. Software organizations need to focus on this area to get long-term performance of secure software with low maintenance cost. Therefore, developers need to focus

on secure as well as durable software. The study will help developers to improve the security for long life span. Further, the technique may be helpful for assessment in other areas.

TABLE OF CONTENTS

LIST OF THE FIGURES

LIST OF THE TABLES

CHAPTER - I
PROBLEM DEFINITION

1.1 Introduction

Security specialists are confronting various issues to comprehend the new security challenges at the beginning periods of software development. There is ceaseless weight on developers to maximize the development and at the same time lessen the expense and time acquired on security to enhance financial performance. The nature of development is ending up more perplexing step by step and requirement for security is expanding in each field. Evaluating and looking after Confidentiality, Integrity and Availability (CIA) amid phases of programming advancement is ended up being extraordinary compared to other approaches to get more secure software [1-2]. Security in the product must be consolidated in software development advancement from the earliest starting point and it ought to be proceeded till the software is being used [3-4]. Consolidating security amid security improvement prompts reduction of development budget and effort. It must not be forgotten by security specialists when advancement of software security development is finished or it ought not be dealt with late amid software development.

According to a technical report, Software-as-a-Service (SaaS) operations and Management Company, about 73 percent of organizations expect to shift nearly all of their applications to Software as a Service by 2020 and want to improve the life-span of services [5]. Almost same is said in report of Cisco’s Global Cloud Index for the period 2013-2018, 59% of all cloud workflows will be delivered as SaaS by the end of this year and they spent lots of time and money to improve the life span of software [6]. Organization of CA Veracode tested a scan of 400,000 numbers on their clients' software in a one-year period which started in April 2016 [7]. In these scans they found 12.8 million flaws. According to the report, it was found that stakeholders who use antivirus software to scan the improvements of security find atleast one vulnerability on the initial scan and thereby enhance security services. About one in eight finds high or very high severity vulnerability related to long life security services. Later on, through these scans the image of software security still remains unclear and uglier. In 2016, companies closed only 58% of vulnerabilities in the same calendar year in which they were found. And the percentage of companies that successfully passed checks for weaknesses on the OWASP Top

10 list declined to 35% for internally developed software, down from 39% in last year's report. Third-party code, which typically has more vulnerabilities, also performed worse year over year: only 23% passed the OWASP Top 10 check, down from 25% the previous year. Globally, the data shows that organizations are trying hard to stay away from vulnerabilities and doing the security checks on a regular basis. Yet there is something missing, still secure software for a long time seems to be a daydream.

The demand for security, like safety and other software quality attributes is growing day by day [2]. As the customer's priority has drifted towards security along with other quality attributes, the developers are also focussing towards the same and striving to develop secure software. As information about an organization's assets is processed through software, security concerns for software grow more for the organizations. To appraise security and to enhance it, organizations need to recognize and address the diverse sorts of security attributes which influence security service life span specifically or in a roundabout way [3]. Developing security might be upgraded by incorporating other quality characteristics into the current properties, for example, CIA. In addition, organizations need to upgrade security to enhance the working life of software security [4]. Accomplishing security of software for significant period of time (duration) isn't a simple methodology. It comprises of significant exercises to be locked in by engineers and additionally analysts. Enhancing security is one of the exceptionally imperative strides towards long life expectancy of security at an early level of software development [5]. With the foregoing discussion and even looking into the problem of security, the following research is borne out to attain the objectives of the study.

It has been observed that the total time of software maintenance is greater than its development, such as 80 % of the development [8]. There is a big question as why does software need maintenance? Unlike physical products, software as well as its security exists only in digital form, which means that it is not subject to wear or decay. So, in theory, it is possible to have a piece of software and security running for years without modification. In practice, this usually doesn't happen. Software and its security are like biological species which have to adapt themselves according to the changes in the environment.

Adaptive maintenance helps in accommodating the changes made to the security [8]. This is helpful, but if organizations will spend more time in maintaining the software than in development, then the concern of security will deviate from development to maintenance. Thus, there is a need of concern over this issue and see the ways to deploy software with security

which does not demand more maintenance time and monetary value [9]. Concerning over this issue reduces cost and time incurred over maintenance.

1.2 Software Security

In the present scenario, dependency on software is so high that life cannot be imagined without them. With the overall advantages of software and the security design on them, there is just about fear as well. Fear of being insecure, fear of being hacked is always there and many more. Consideration for software security during development, thus emerges as a helpful solution to the user. Software security is a branch of software engineering, which aims at preventing security problems by building software without security holes [9]. According to G. McGraw, software security is about building secure software, i.e. designing software to be secure, making sure that the software is secure, and educating software developers and architects, and users about how to build secure software [4]. Due to the wide applicability of information systems, software security has become a crucial component of every software engineering process. Software security is one of the most significant features in software evolution process that calls for high attention among engineers. Indeed, software faces threats from various potential malicious adversaries that are raising every day; from web mindful applications running on PCs, to complex media communications [4, 10]. These threats can impose a vast challenge to software engineers in planning measures as a portion of their risk management activities as well as, designing the appropriate security requirements and policies. This is due to the degree of subjectivity in how security is being perceived and subject to different levels of concerns. Moreover, numerous software is developed without paying due attention to security issues including confidentiality, integrity, access control and non-repudiation.

1.3 Software Durability

Quality is a noteworthy element of the software which tends to improve durability [11-12]. Durability is an important issue in evaluating the software quality. Development of software design is not a one-time built-in process; it is based on the reuse of existing specifications in the market. Analysis of service life of the secure software relating to quality is the key point of this research work. Usually, software is delivered without considerable security, which welcomes vulnerabilities. To mitigate them patching is done, which further results in more vulnerabilities. It is normally expected that the design will remain serviceable for the whole life of the software and that services and qualities may come and go [12-13]. This leaves designers and users to

consider the relationship of the durability to the rest of the software architecture. Software durability is a term used to describe the usefulness of service-life of software with optimal maintenance.

There has been a lot of work done in the field of software maintenance in regard to durability as Nathan Ensmenger in his article 'When good software goes bad: the surprising durability of an ephemeral technology' talks about problems in maintenance [13]. This work also stated that there is a need to focus more on problems related to maintenance achievement. It is stated that software durability is related to software serviceability, and it has been pointed out that achieving durability may enhance the serviceability of the software. Software durability is defined as "Service of a software product is durable, if it works efficiently, effectively for user's satisfaction up to his/her expected duration" [12].

1.4 Security Durability

The fundamental point of the innovation improvement is to serve mankind concerning social up degree and secure client from malignant assaults. At each phase of development life cycle, paying attention to the security of the software may increase high reliability and user satisfaction [14]. Security of software effectively increases the quality to meet its business requirements. Security experts say that process of identification of security factors is carried out at the time of security evaluation. Practitioners need to concentrate on security during an early stage of software development; however, it is not hundred percent achievable [15]. Longer security during software development is now becoming a difficult task for the security developers [9]. Also, it needs consideration of security which includes security attributes, classifications, and security measurements. Security attributes must be considered as an important tool in every level of software development. Security attributes are identified as an incredibly vital surrounding in security engineering. Identification of security attributes helps to improve security during software development [16]. These attributes prepare a core part in the security world. Further, security attributes are included to produce solid cryptographic arrangements, as well as to discover an approach to give security necessities to enhance security amid software development life cycle [16-17].

Software security affects the duration of the service life of software [11-12]. This statement fortifies the fact that there must be an attribute which relates to security i.e. durability. In this concern, durability should be considered as one of the supporting attributes of security.

Durability, in terms of software is the time period during which software gives services [18]. Henceforth it appears that security highlight has changed to the durability of the software. Security is straightforwardly or by implication associated with the administration life of the product. Durability is further directly or indirectly involved in the security of software and vice-versa [18-19]. This research is dedicated to conclude theoretical and empirical facts of security durability through assessment. Commercial software products typically have a kind of visible features, which further give a market advantage over security, such as the CIA and similar less evident attributes [20]. The main objective of this contribution is to reduce the efforts to manage and control security. Assessment of security durability may be helpful to improve security and optimal maintenance for a period.

1.4.1 Needs and Importance of Security Durability

The importance of services provided by the software is equally needed as software itself [21]. Long runtime services are preferred over in software security. These services must also assure integrity of the information processed through software. The importance of long run services and security is being recognized by security developers, that is why it is becoming one of the necessary requirements during software development [22]. In many areas of business, such as finance and accounting, business processes are quite identical and secure [23]. They don't differ considerably across organizations, and they used to be stable on time. Hence, to survive in the competitive global environment, software development organizations must focus on security services during software is in use. In addition, with the increasing need of longer and high security of software, developers are pressurized to focus on the software whose security is durable (long life-span) [24-28].

Durability as a pillar of security is considered to maintain CIA for long time period [18, 29]. This research will come out with the current challenges faced while ensuring security durability, and proposes a way forward in terms of security. The quality of software also depends on security of software [4]. Durability is an issue that can be approached from multiple viewpoints, that is why many different disciplines, such as psychology, civil engineering, and sociology, are trying to tackle it [30]. Achieving longer security services is becoming a crucial event for the development organizations. In this concern, there are so many reasons to focus on longer service life of software security and some of the pertinent reasons are as follows:

- A corporate structuring software policy ought to be utilized, and a legitimate documentation should be inferred [23].
- The security durability assessment policy during development life cycle of the software should be defined clearly [24].
- Guidelines, processes, and suggestions need to be given for supporting the security services for improving the working life of security as well as software [25].
- Security teams for secure designing need to work with software development teams to incorporate durability concept within the various development styles that are being used [26].
- Appropriate tools should be used for security assurance of durable software [27].
- The policy, guidelines, processes, checklists, infrastructure should be updated at regular intervals to accommodate the user needs and technology changes [28].

From the above discussion, it can be concluded that improving the working life of secure software is going to be a new challenge for the software industry. Further, with the help of security durability assessment, a service life of security as well as software should be improved.

1.5 Multi Criteria Decision Analysis

Software security assessment seems to have different types of criteria within it. Such as to assess software security, one needs to assess its attributes like confidentiality, integrity, availability, authentication etc. Security durability has multiple attributes that are thoroughly defined in the next chapters. Hence assessment of security durability is also a multiple criteria problem. To assess security durability with its contributing attributes, multi criteria decision making techniques will be used in this research work. Multiple Criteria Decision Analysis (MCDA) or Multiple Criteria Decision Making (MCDM) is one of the most important methods for assessment with multi criteria having multiple levels [31]. A brief introduction of MCDA methodology and its associated techniques are as follows:

MCDA methodology helps in making decisions among the multiple conflicting criteria [32]. In daily life multiple criteria problems can be solved using MCDA methods such as a selection of one criterion from different criteria [31-32]. Research based on MCDA methods has a history of

only few years. Further, usage of the internet in everyday life has created a number of problems of multiple criteria [33]. A MCDA problem is generally described using a decision matrix [32]. Suppose there are *s* alternatives to be assessed based on *t* attributes, a decision matrix is a $s \times t$ matrix with each element Y_{ij} being the *j*-th attribute value of the *i*-th alternative.

So far in the field of information technology, MCDA methods have been applied for many purposes including information security, network security, computer security [34-37], but there has been no work done on decision models applied to assess security durability of software. There exist a wide variety of decision-making models, but a selection of single method or combination of methods is a challenging task which mainly depends on the type of decision problem. In this work, researcher uses multiple criteria decision-making method for choosing the appropriate attributes of security durability and assessing the security durability of software. To solve this, multiple criteria decision-making problem, an extensive literature study was carried out by the researcher on available models in the next chapter.

1.6 Pertinent Issues

The security problem arises due to the lack of inherent security measures. Measuring security is about using common sense. Considerable efforts have been made to ensure security by the researchers and industry professionals. But it may generally suffer from delayed security assessments, which counts heavily towards security and quality assurance measures [38-39]. The effort in respect of early and accurate security estimation needs to be undertaken for worthwhile, software development. It appears inevitable to have a potentially effective approach for an early, on time and accurate estimation of security durability during the software development life cycle. At early stage of development life cycle, it is mandatory to determine what to measure, organize the variables in a way that makes them manageable and meaningful, and build security durability that works efficiently for longer services. It is trusted and well accepted that security must be integrated in the software from very early in the software development life cycle as soon as the design starts.

The software is neither hundred percent secure nor it can be [40]. There might be some identified security flaws present that were not fixed during software development due to time constraints or any other reason [41]. These flaws are looking again, prioritized and fixed. Further, maintenance is an ongoing process and does not end until the software is completely out of use or taken over by a new software. Due to high security maintenance cost and time,

there need to optimize it (maintenance cost and time) [42], and also, increment in the security life span of software services is needed [43]. With the help of long-life span of security, software is durable and profitable for the organizations. Integrating security durability during early stage of software development leads to optimization of the security maintenance cost, time and effort [12, 44]. Based on the description of the above problem there may be a vast set of research questions that may need to be addressed. Some of the pertinent ones are stated as follows:

- Are there any problems in the way of organizations to perceiving software security?
- What are those problems?
- Are these problems affecting the security maintenance and life span of security?
- How to minimize or optimize the security maintenance cost and security maintenance time for improving life span of security as well as software services?
- What are the factors that directly influence the security of software?
- What are the factors affecting durability of software?
- Is there any relation between security and durability?
- How can we relate to security with durability?
- Is there any standard mechanism available for assessment of security durability?
- Is it possible to estimate security durability at early stage of software development?
- Can we get a mechanism, which may be used in early stages of the software development life cycle to estimate the security durability successfully?
- Can we develop a security quantification model targeting durability?
- How can we improve the software life span through security durability estimation?
- How can we improve the estimated security using developed models and guidelines/suggestions?
- What should organizations do in order to develop secure as well as durable software?

1.7 Problem Formulation

In these days, practitioners try to design software to be secure and longer in use, because maintenance cost of security has invariably increased [45-46, 146]. Now, the question arises, how to optimize maintenance with improved security to face upcoming challenges related to new security issues [42, 45]. To control and manage security within a life span of working software, the developer's focus should be on design phase rather than maintenance, because, design is called skeleton or blue print of software. It is comparatively easy to integrate security at this phase. At the same time, if security aspects are ignored in this phase, the resulting design might become vulnerable. Even a single flaw in design will manifest in the next phases of software life cycle and it may become difficult to diagnose the risks as life cycle proceeds. On the contrary, reducing design weaknesses reduces rework and cost for the further phases [47].

Nowadays, security durability in software is in demand [18, 44]. Further, the software that is secure, trustworthy and usable for a long run is considered as durable. By leveraging its unique expertise in the performance of software security, this thesis offers integrated solutions and engineering guidelines for the security durability software development. This problem requires us to define what we mean by longer security/ security durability. In addition, there is a need of relationship between security and durability through the attributes [18]. To assess security durability, there needs to be a prioritization of security durability attributes to design security effectively and enhance the security durability. With the help of the prioritized security durability attributes through the decision-making methods, security designers come up to a focused on secure as well as durable software services during the development process [48-49].

Many promising techniques have been grown, unifying security attributes, security durability, and security functions, although, it is difficult to estimate the security durability at early stages of development. Quantitative analysis at early stage enables the evaluation and assessment of security durability. It provides the basis for assessment of security durability. Assessment of security durability will help to dissolve trade-offs between security goals and maintenance cost or rework [50]. Assessment is the valuable technique of understanding for improving, guiding, and controlling security durability integration at the early stage of development process [51]. A preliminary investigation of the literature reveals that even though there has been plenty of security problems fixed, still the produced software is not secure for a longer duration [52]. Some of the issues in this regard are listed as follows:

- There is a need of a mechanism to maintain CIA for a specific duration.
- There is a need of mechanism to optimize security maintenance process.
- There is need of a mechanism to enhance duration of security of software design during development process.
- There is a need to bridge the gap between security attributes and durability attributes.
- There is need of security durability development framework that is a challenging task and needs an in-depth analysis.
- There is need of a mechanism for the assessment of security durability.
- There is need of a mechanism to understand the user's expectations towards security durability.
- There is need to bridge the gap between security durability consideration and its actual implementation.
- There is need of a mechanism to help in better security monitoring and controlling of software under development in a life span.
- There is need of a mechanism to reduce efforts in producing secure and durable software.

From the foregoing discussion, it is apparent that eliminating unwanted maintenance cost, time and efforts to the development of secure software with improved working life span is trending topic to be discussed. There is not even a single full-proof mechanism available for addressing security durability. A viable assessment model is needed to address design security durability. From the proposals of the forgoing references it is evident that early availability and use of quantitative measures of security is a key factor to a successful delivery of secure software [53]. Therefore, there is need to develop a mechanism to quantitatively estimate security durability during development life cycle. A study on development of a mechanism to quantify security durability is proposed under the aegis of the problem entitled:

'Fuzzy Multi Criteria Decision Analysis for Security Durability Assessment'

1.8 Research Objectives

In today's world, the use of secure software is increasing rapidly for business, educational, commercial purposes, etc. As per the increase in the use of software, complexity of security is also increasing. And with the increasing complexity of secure software, developers have to contribute more for fulfilling customer's requirements [21]. Despite all the efforts, software is still not secure enough. Over the years, the level of threats to software has varied depending upon the many factors as the environment in which software is used after development is not under control. The organizations spend lots of money to solve security related challenges during software development. In addition, organizations want to enhance security to improve the working life of software. Software security assurance is not an easy process for a longer life span [19]. It consists of some necessary steps to be taken by developers while developing the software security in the early stages of Software Development Life Cycle (SDLC). Improving durability of security is one of such an important step towards improving security of software. In order to improve security of the software, developers need to improve the performance of security factors including confidentiality, integrity and availability during software development. Hence, there is need to focus on increasing secure life span of software.

Identification and remediation of security threats after deployment can be a time-consuming and costly exercise. It is far easier to build a durable security design than to fix an insecure one [18]. Though, durability is essential for secure software services, it is hard to achieve while designing security during software development [38, 147]. At its core, the value of secure software is derived not only from its secure services to increase productivity and efficiency, but also from its resilience to attack, always performing at required level during its working. Developers may enhance the duration of security during the working life of software with less probability of security failure [39, 148]. This research is focused on increasing service life of software through security durability assessment. Following are the objectives to be followed when building a roadmap to security durability of software:

- To develop a security durability program plan that includes other key factors
- To identify and investigate their potential design objectives
- To identify and investigate the durable security necessity of serviceability of software, an arrangement as a benchmark for quality

- To identify new attributes and their relation with durability to provide service life of secure software for a specific duration
- To develop a viable and prescriptive framework for security durability estimation using its properties
- To assess the security durability parameters using this developed framework
- To validate and test the proposed framework

Given the security related objectives, it is highly in demand to make an effort to produce secure as well as durable software. One of the pertinent ways is integrating security durability within the development life cycle. Insuring life span of security early in the development process may reduce costs and rework during secure and durable development. Under the aegis of this project, it is aimed to explore the possibilities for developing a methodology to estimate security durability at an early stage of software development life cycle in order to optimize the security assurance effort at overall level for a specific life span. The basic idea is to assess security durability during software development life cycle and fix the problems at the earliest without any delay.

1.9 Research Methodology

Security of software with poor durability is likely to fail in a highly competitive market; therefore, developing organizations requires more attention towards ensuring the security durability of their software [54]. To be able to develop durable as well as secure software cost-effectively, there is a need to assess the security durability during software development process.

Organizations need to focus on this area to get long-term performance of secure software with low maintenance cost. Thus, developers must focus on security durability during development process. This will assist in developing life span of security of durable software which is likewise a major requirement of users. Objective of the proposed research work is to enhance the life span of security, thus to improve the service life of software. The methodology is supposed to accomplish through several phases including following:

- Conceptualization, Review and Requisite Specifications
- Development of Framework

- Development of Phased Security Test Process
- Expert-Review and Revision
- Implementation-Level Specifications
- Implementation, Preview and Pre-Tryout
- Assessment of Effectiveness
- Documentation and Finalization

In this research, we will be focusing on assessment of security durability. The results will help in improving the working life of secure software. In addition, the results will help in decreasing the maintenance issues (like cost and time) for serviceable software.

1.10 Significance of the Study

A series of tragedies and chaos caused by the insecure software proves that the duration of software security may be simply a matter of life and death at time. Software industries are now focusing on longer security services of software as a major concern [55]. Software security measurement and improvement has been one of most talk about topics in organizations. In addition, identifying and addressing various security attributes during software development may reduce maintenance time and costs [56]. Security durability may be considered as one of the supporting attributes of security. Because, durability strengthens the fact that longer security doesn't need maintenance for a specific duration [12]. This decreases the cost and time incurred in maintenance. Security durability assessment may intensely influence security of the software.

The investigation of security durability parameters and their effect on security will ease up to reveal the qualities and shortcomings of the security strength. The precise estimation of security durability remains a vital issue in light of the fact that there is supposedly no great comprehension of the idea of security durability. There is no unmistakable definition to 'what perspectives are identified with security durability'. Finding an appropriate method to measure security durability and the greater part of the angles identified with it is exceptionally troublesome. Hence, an examination on security durability assessment winds up vital for security developers, programming engineers and their clients. Durability applies a methodology that conveys robust, vibrant security to support, facilitate all business initiatives, including

clouds, mobility, and improve security. The main advantages of security durability assessment are given below:

- Improved probability of life time of security software
- Reduced cost of maintenance on security development life cycle
- Reduced maintenance and repair costs of software security
- Improved satisfaction of user's and market value of the product
- Prioritized security durability attributes and guidelines may be helpful to design secure as well as durable software
- Field of security is still in its infancy and only quantitative assessment of security durability may facilitate the mechanism on predicting how long the software is secured.
- Since quantitative assessment techniques for security durability are not available, the security community primarily uses qualitative assessment techniques for security. The proposed study may help industry professionals in producing a quantitative estimation of security durability.

A consistent quantitative estimate of security durability is highly desirable for secure software during development life cycle. The literature survey reveals that nothing significant, precise and clear exists in this regard that can be used to quantify security durability in early stage of development. Therefore, in absence of any framework or model for quantifying security durability, it is worthwhile developing a methodology for security durability quantification. The main aim of this research is to gain an in-depth understanding of the durable security/security durability concept and the need to design durable as well as secure software.

1.11 Limitations and Delimitations

Every coin has two sides. In research point of view both surfaces hold imperative position. However, the positive appearance offers new dimensions to the proposed study while the negative portion highlights the deficiencies of work. After resolving the deficiencies of the intended work, the redesign efforts ascertain innovative feature of lessons. Despite having so many reasons favorable for the industrial adaptation of the approach, there are negative aspects also. Some pertinent ones are listed as follows:

- The approach can be applied to measure the security durability only for specific versions.
- Due to unavailability of big industry data, the proposed framework is only validated with a small set of data.
- To focus more attention on security durability quantification area, only a set of security factors and durability factors are chosen amongst various security factors and durability factors respectively.

The recognition of security durability assessment information is based on perception.

1.12 Thesis Outline

A thesis of the research has been prepared to reflect the detailed study on the research problem and aforementioned research questions. The thesis consists of eight chapters. Following is the chapter wise summary of the research thesis.

Chapter-2: Literature Review

This chapter is associated with the existing framework/ approaches/ metrics/ tools/ methodologies for enhancing/ improving service life span of security and of the software. A thorough review on the beginning of the security measures era to till date is presented. The expert's view on durability/ security quantification and minimization are explored. The detailed review of some important existing approaches (related to security, durability, security durability and MCDA) from the year 2003 to 2017 is presented. On the basis of the review, it is concluded that assessment of security durability is needed to improve security life span of software services and maintenance cost and time minimization is much advocated but done least. Also, after literature review it is clear that unfortunately, a proper framework for assessing security durability is missing.

Chapter-3: Software Security Durability Assessment

This chapter discusses the effects of durability on security of software by explaining the effect of different attributes on the security durability. This chapter identifies the characteristics of security and durability attributes. Further, the identified durability attributes are verified against identified security attributes and mapping of these attributes through the classification at level 1, level 2, and level 3 is done. Further, step wise mechanism of security durability assessment is given in this chapter.

Chapter-4: Development of Security Durability Assessment Framework

This chapter discusses how security of software can be enhanced by improving the security durability. The premises of the framework are set. A framework for security durability assessment is developed. The framework encompasses complete guidelines for identifying security and durability needs in the industries and creates the suggestions/ guidelines for improving the security durability of software.

Chapter-5: Implementation of the Framework -Using Fuzzy Multi Criteria Decision Analysis-

This chapter implements the framework developed in chapter 4. Level wise attributes of security durability are identified in chapter 3 and step by step, implementation process also discussed in chapter 3. According to the implementation of the framework, firstly, researcher is using one of the most famous MCDA techniques which are called fuzzy analytic hierarchy process to evaluate weights of the attributes. With the help of these weights, researcher categorizes most important attributes at each level and proposed some suggestions to improve the life span of security of software. To evaluate the ratings of the attributes of security durability, two successive version of a case study has been taken. i.e. entrance examination software for Babasaheb Bhimrao Ambedkar University, Lucknow, India (BBAU Software). With the help of weights and ratings of the attributes, researcher estimates the security durability. Initially, security durability of the BBAU Software first version is assessed. After the suggestions, ratings are evaluated again and then security durability is again assessed of the BBAU Software second version.

Chapter-6: Implementation of the Framework -Using Classical Multi Criteria Decision Analysis-

After implementation of the framework through fuzzy multi criteria decision analysis in chapter 5, this chapter is implementing the framework again through classical multi criteria decision analysis. Main aim of this chapter is to prove the correctness of the approach used in chapter 5. After the assessment through classical method, researcher assesses the difference between both calculations statistically.

Chapter-7: Experimental Validation

Improvements in the results after the suggestions/guidelines are given in this chapter 5. Sensitivity analysis of the results is shown in this chapter. Further, statistical study is carried out to show the acceptability of the framework. Also, this chapter presents theoretical and empirical validation of the framework proposed in chapter 4. Theoretical validation is done on the basis of the experts' feedback for the approach and statistical validation is also done in this chapter.

Chapter-8: Summary and Conclusions

This is the concluding chapter of the thesis. In this chapter, major research findings along with the other findings are presented. The research questions posed in the first chapter are addressed one by one. The significance of the research is discussed in details. Future plans for extending the study are also discussed.

CHAPTER - II
LITERATURE REVIEW

2.1 Introduction

At present, in order to gain profit from market and fulfill the needs of secure software, security durability will play an impactful role. To maximize security durability, the role of maintenance is crucially important. The challenge is to discover “what works today related to optimal maintenance of security and improve the life span of software services” from the practical and theoretical point of views. It can be seen that maintenance is 60-80% of total cost and development is at most 20% of software life [55]. As most of the companies today doesn't seem to acknowledge they mostly focus on faster development and set due dates without proper estimation of security. This forces developers to dump and go, which subsequently makes the maintenance harder.

It is a challenge to identify the changes in design for improving the security and turning it into optimal maintenance of secure software product. The main causes of software security failure are: the insufficient involvement of security designers during software development process; the fact that the security needs were incompletely defined; experts are always focusing on maintenance process rather than design to optimize maintenance and improve security for a better life span of software. To enhance working life/ service life of software for better understanding of maintenance issues, the practitioners must focus on the software design and its durability.

Without it, the practitioners may not optimize the maintenance issues of a profitable and successful software product. To optimize the maintenance issues, security durability plays a key role. Further, security durability and its attributes are always ambiguous and have multiple meanings; their description is often linguistic and vague. Moreover, it is recognized that expert’s judgment is always imprecise. The usage of the fuzzy logic is recommended to model the uncertainties of expert’s preferences [57]. To enhance the security for a life span, there is no framework/ method available for security durability assessment and improvement during software development process. To understand the relationship between software security, software durability and expert’s judgment, there is a need for a critical review of literature. This chapter will focus on the literature related to security, durability and MCDM techniques used

related to security of software. Moreover, this chapter will also focus on the security durability which are related to other fields including computer hardware, data storage etc.

2.2 Literature Review on Assessment and Improvement of Software Security

Software has been involved in everyone's life in various forms such as to share data, to communicate, to maintain databases etc. Almost every field of life is connected through some kind of software as medical, engineering, social and others. All information related to software must be secure. Consequently, demand for secure software has increased today. Software security can be termed as the idea to secure software from malicious attacks and fraudulent persons or hackers [58]. Many experts have discussed many areas of security including security attributes, security management, security maintenance etc., but still there is something missing. Development organizations spend their money and efforts to optimize the maintenance of security for improving the life span of the software [59]. But, they are not yet successful. Some of the pertinent efforts of the practitioners to assess and improve security of software are given in table 2.2 (a).

Table 2.2 (a): Pertinent Work on Security Assessment and Improvement

S. No.	Authors/ Years/ (References)	Contribution Type	Focus (Threats/ Goals/ Problems)	Assessment and Improvement (Single level/ Multilevel)	Problem Addressed	Summary of the Contribution
1	Davoud Mougouei 2017 [60]	Fuzzy Inference System	Security Goals	Multi Level	Modeling process is defined through Implementation.	Author defined the problem in existing prioritization techniques for security attributes and the needs of prioritization of attributes. These are usually ignored and thus give birth to new but insecure software. To address it, the author proposed considering partial satisfaction of security needs when tolerated rather than ignoring those security needs for the future. As a result, this research has contributed a framework that prioritizes and selects security requirements. The proposed framework takes the security model as the input and uses fuzzy inference system to use linguistic terms.
2	Robert E. Crossler 2014 [14]	Multi Dimensional Scaling	Security Threat	Multi Level	Author examined three security threats.	Author discussed the quest for complete security in user's point of view. The study has taken 279 individuals' security behaviors and analyzed them using multi-dimensional scaling. In addition, he examined three security threats after analysis: security related performance degradation, identify theft, and data loss. The results were presented as a mapping of security behaviors that performed together. The data were collected using expert reviews for the different dimensions. The work advises to research and practice security

						by identifying security threat-response pairs via expert interviews, surveying individuals.
3	Friedrich Praus et al. 2016 [15]	Survey Based Report	Security Goals	Single Level	Critically examined the security and software architecture.	Authors presented a research on software security requirements in building automation. This paper provided an extensive survey of the security requirements for distributed control applications and analyzed software protection methods. Architecture on the same problem has been defined that works on to secure software that runs on different devices or classes. This architecture also prevents attacks on smart homes and buildings.
4	Sarah Vonnegut [20] 2016	Survey Based Report	Security Problems	Single Level	Main reasons of security failures are addressed.	Author discussed about the software security assurance program. According to him developing secure software should be an essential part of any organization. For a strong security assurance program, he has given four key steps, including a strong focus on security awareness & education, established secure development practices and procedures, automated security testing for a secure software development lifecycle, and ongoing security assurance program.
5	A. Purdy, 2016 [16]	Technical Report	Security Problems	Multi Level	Measurement of Human efforts, cost and sustainability is addressed.	The authors gave a report related to cyber security. Cyber security research and development strategic plan defines "sustainability" of secure systems development, as meaning in "cost-effective". If it is true, then developing critical and advance systems is cost effective. The key concern of developing secure software systems without measuring its sustainability is quite impossible; hence the measurement of human efforts, cost and sustainability are important.
6	D. Gray et.al 2015 [17]	Technical Report	Security Problems	Single Level	To manage the cyber related issues, a framework is proposed.	Authors proposed a framework to improve federal cyber security governance through data driven decision making and execution. This report analyses that the federal government must make better enterprise-level cyber security decisions in the shortest time possible. The report also observes and decides the ways used by the U.S. Department of Defense to enable decision makers at the strategic levels of government to best make the decisions for the success while development of the system. The report also discussed the difference between cyber security governance and cyber security operations. Finally, the report discussed key considerations to ensure success at the point of execution based on work performed in the Observe, Orient, and Decide phases of the Observe, Orient, Decide, Act loop.
7	Robert Evans 2015 [61]	Technical Report cum Dissertation	Security Problems	Single Level	Software security assurance program need to be	Author submitted a dissertation on integrating security into the undergraduate software engineering curriculum. The report describes the existing software assurance program, also called software security knowledge, its methodologies and information security

					incorporated in development life cycle has been discussed.	technologies that are currently being recommended by stakeholders. The software security assurance program needs to be strictly incorporated while developing. The report concludes with the facts like software security assurance program of software engineering needs to evolve to where it recognizes the necessity to produce software that is free from vulnerabilities.
8	National Institute of Standards and Technology (NIST) 2014 [21]	Technical Report	Security Problems	Single Level	Guidelines are provided	Author presented the report of building effective assessment plan. The report provides a set of procedures of security and privacy controls for conducting assessments. The procedures are implemented in various phases of development life cycle. It is customizable. Further, it supports risk management processes. The results also provide the guidelines to analyze and assess security as well as privacy during the development life cycle of software systems.
9	P.K. Chouhan et al. 2015 [22]	Theoretical Analysis	Security Problems	Multi Level	Analysis of SAAS in four different perspectives is done.	Authors presented a research paper in which they defined software as a service (SAAS) in different perspectives. Author defined SAAS in four different perspectives such as Software as a service in cloud computing, software as a service in mobile cloud computing, software as a service in software defined networks and software as a service in Internet of Things. In addition, author also described security challenges in software as a service. According to the work there are three types of security challenges that are data security, application security and deployment security. Analysis of security in software as a service is also done in this work and the results were discussed.
10	Michael Hoehl 2013 [62]	Research Paper	Security Problems	Multi Level	Framework for security patch management is given.	Author proposed a framework for the security patch management program. The framework has several steps, including vendor notification, tracking, risk assessment, packaging, and deployment. The framework was based on authoritative standards. The patch management process is similar to the defect management process during software development life cycle. The framework could be properly managed by using best practices. It was proposed to be very useful at every stage of the development process, including policy, risk management, standardization, asset inventory control, and metrics
11	Kakali Chatterjee et al. 2013, [63]	Research Paper	Security Goals and Threats	Multi Level	Framework for early integration of security in software development	The authors proposed a framework for the development of secure software. In the author's view to design, build and deploy secure systems, it must integrate security into the software development life cycle. The author suggests that developers should adopt best practices and methodologies to include specific security-related activities. This paper discussed that practitioners usually apply security measures and plans at the earliest stage of

					process is given.	development, but these measures usually fail in the late stages of SDLC. The author identified a framework which eliminates this problem by engaging security requirements in the all stages of development. The identified design attributes affecting security requirements were also prioritized and a security design template was prepared by this research.
12	Regina Thienne Colombo et. al. 2012 [34]	Analytic Hierarchy process	Security Goals	Multi Level	Framework for attributes prioritization is given.	Authors proposed a framework for prioritization of software security intangible attributes. The report presented a theoretical framework based on mathematical constructs to score the priority and to estimate measures of security factors. The framework helps in converting the complex system decomposition into simpler and smaller systems. The report could also be used to specify security needs. In addition, the framework helped to understand and measure the security attributes. The results provided the priority score of security factors that is calculated from analytical hierarchy process and helpful in optimizing maintenance.
13	S. A. Khan 2012 [64]	Model-driven methodology	Security Goals	Single Level	Proposed a development framework to enhance security through assessment.	Author in his research raised the issue of security problems due to design complexities. The author described that cost and time to inbuilt security in software at the last stage of development is high. Hence this research focused on securing software during the design of software. Design complexity invites bugs and so it is advised in this paper to consider complexity attributes in design to improve the security of software. In this work author identifies security complexity factors in perspective to its impact on object-oriented factors. It provides a development framework which incorporates design complexity attributes into the development to enhance security. This framework includes premises and generic guidelines to be followed for securing the software. The framework is then validated by quantitative study of an experiment and given three models consisting confidentiality quantification model, integrity quantification model and availability quantification model.
14	Alka Agarwal et al. 2011 [65]	Model driven methodology	Security Problems	Single Level	Proposed a framework for vulnerability minimization.	Author in one of her works identified that the security issues are rising because of the vulnerability flaw in the design of software. Hence, to improve the design of software author presented a framework which works on object-oriented design and resolve the issues of security which are encountered mostly due to vulnerable design. The framework presented here identifies the attributes of object-oriented design flaws, analyze it and propose security metrics. This security metric is helpful in the development of secure software. The thorough literature review of the above research points the fact that security is a major concern in every area these days. It also strengthens the fact that design plays an important role in assuring the security of software. Complexity

						of design and attributes such as confidentiality, integrity and availability play a foremost role in software security assurance.
15	Smriti Jain et al. 2011 [66]	Technical Report	Security Problems	Single Level	Discussed on Security Breaches.	Authors presented a report on security breaches that are usually caused by the vulnerable software. The report described software security requirements gathering tools that help to gather security requirements from the various stakeholders. These tools help the developers to gather security requirements along with the functional requirements and also incorporate security during other phases of SDLC. The report also discussed the scenario of using these tools with the software requirements specification document as specified in standard IEEE 830-1998. The results provided here by the case study is detailed which further includes web enabled software as highest requirement for security.
16	M.T. Dlamini 2009 [38]	Survey based Report	Security Problems	Single Level	Reviewed previous work on security issues.	Author in his work defines information security as not about looking at the past with anger of an attack once faced, but alert for the organizations to secure themselves from the next attack. The author also provided the results of a survey of the industrial reports and review of the past publications done on this topic. The paper highlights the important information security issues that were not addressed yet. The suggestions were also made by author that new research should be conducted to overlook the issues uncovered in his research.
17	Alvaro A. C´ardenas et al. 2009 [39]	Technical Report	Security Problems	Multi Level	Critically examined the security challenges.	Author in his work signifies the challenges in securing cyber physical systems. In this research author discussed the challenges such as understanding threats, identifying the properties of these systems and apply the security features or mechanisms to avoid the security breaches. The paper describes some potential threats to any cyber systems such as cyber criminals, dissatisfied employees from any organization, terrorists and criminal groups, etc. This work gives a proper difference between corporate IT security and cyber physical systems IT security. It raises issues of legacy designs being used in present systems, which are a great cause of vulnerabilities in any system software.
18	2003 Gary McGraw [4]	Research paper	Security Problems	Single Level	Discussed about secure software development in industry perspective.	This research work discussed software security as a big concern today. According to the author it is a growing problem day by day because of the growing connections of systems. He further adds that software is being complex and hence securing this is a potential issue. Penetrate and patch framework solution is no longer effective in this, he added. This work is engaging software quality, security risk and software security in a single chain and discusses some real-world counter measures in security.

In the table 2.2(a), the most essential researches have been reviewed and examined. The problems are addressed through the researchers. From this part of literature review, researchers summarized the different researcher's work and found this that along with fixing security issues; design of security should also be strong. Hence to improve the security, designing is the main point during secure software development. As the new threats are coming in the future, new security issues are generating day by day and for fixing these latest security issues maintenance cost and time is spent more on security development. To reduce these cost and time there is a continuous pressure from user end. Many researchers are trying to fill the hole of security design so that new threats are removed and security services are enhanced with it. To improve the software's service life, security life span should be improved. Further, literature review on durable software services is discussed in next portion.

2.3 Literature Review on Durable Software Services

There are three important service quality thresholds associated with security durability: 1) the quantified quality, recognized by the software developer or minimum codes 2) the minimum acceptable quality indicating the need for replacement; and 3) failure. Though, there is a little work available in the field of software durability. Unfortunately, no literature is seen focusing on security durability assessment of software. Some researchers have defined durability in terms of trustworthiness; while the other has defined in terms of dependability and human trust. Some of the pertinent work related to durability has been shown in table 2.3 (a).

Table 2.3 (a): Pertinent Work on Software Durability

S. No.	Authors/ Years/ (References)	Contribution Type	Focus (Threats/ Goals/ Problems)	Assessment and Improvement (Single level/ Multi level)	Problem Addressed	Summary of the Contribution
1.	Celia Chen 2017 [42]	Research Paper	Maintainability based Problems	Single Level	Maintainability is defined as a big concern for non-durable software.	Author described Why Is It Important to Measure Maintainability and What Are the Best Ways to Do It? According to author high maintenance can cost approximately 75% cost of the whole cost in software development. Moreover author defines software maintainability as the ease with which a software system can be repaired or modified, to correct faults, improve performance or other attributes, or adapt to a changed environment. It discusses that durability of software is improved by reducing the cost and time involved in maintenance. Author discusses that there are metrics that can help software developers to measure and analyze the maintainability of a project objectively. In this research paper, authors addressed the

						importance of understanding software maintainability, gave framework and some of the best ways to measure maintainability.
2.	Security Awareness Program Special Interest Group PCI Security Standards Council 2014 [67]	Technical Report	Problems based on vulnerability reduction and serviceability enhancement	Single Level	Optimal Maintenance is addressed for vulnerability minimization.	Security Awareness Program Special Interest Group PCI Security Standards Council published a special report on the workshop on software measures and metrics to reduce security vulnerabilities. The goal of the report is to gather ideas on how the federal government can identify, improve, package, deliver, or boost the use of software measures, metrics to significantly reduce vulnerabilities and enhance the working life of software with optimal maintainability. The report contains observations and recommendations from the workshop participants. The report includes position statements submitted to the workshop, presentations at the workshop, and related material.
3.	Kelty C., Erickson S. 2015 [11]	Research Report	Software Durability	Single Level	Decaying software durability is addressed with maintainability issues. Design is responsible for less durable software	Author here discussed about achieving durable software with optimal maintenance. According to authors durability of software depends on its different applications such as social, economic and cultural field. Durability is a result of robustness and maintainability. The paper explains maintainability as a never-ending process and hence reduces durability. The experimental results presented by them may prove it to be useful in different perspectives. The article clarifies that bad design is a major reason for lacking durability in software. They suggest to readers for finding the ways of ensuring durability of software by design because it still needs to improve for better user experience.
4.	Nathan Ensmenger 2014 [13]	Research report	Software durability concerned	Single Level	Maintainability plays a key role in decreasing the durability of software. but solution for this problem is not given	Author says that software durability and software serviceability are two faces of the same coin. There is a significant issue of long time services and much cost spent on maintenance of software. Further, the author discusses about working life of durability which decreases as the time passes. Hence, for long-term software, durability is playing a key role. They also related durability with maintenance, as time wasted upon maintenance can be reduced considering the factor of durability in s/w. At the end author concludes that maintenance can be a central issue in the history of software, the history of computing, and the history of technology if do not deliver a durable software
5.	J. J., Cusick 2013 [44]	Research Report	Software Durability and Software Quality	Single Level	Software durability and quality should be maintained	Author defined durable ideas in software engineering in terms of concepts, methods and approaches with the help of virtual

					through the dependability perspective	toolbox. He has mentioned about the need and importance of maintaining the balance between durability and quality during software development process. The author has addressed this issue with respect to durable software.
6.	E. V. Bartlett 2013 [68]	Research report	Maintenance time and cost issues	Single Level	Reliability and durability difference is described.	Author defined the process of maintenance is invariably increasing these days but longer working life of secure software, may decrease the cost of security maintenance process. The author has also suggested the ways to reduce the time spent on maintenance process. According to it durability of software plays an important role in maintaining quality. Paper further concludes that incorrect planned maintenance strategies can actually increase life cycle costs, and reduce reliability and service life of software. Reliability and durability both are important for longer service life with user's satisfactions. After high profile design breaches, practitioners are trying to improve design because it is not enough. Author concludes by differentiating the terms of reliability and durability, as most of the time they are considered same. According to author there is a very small and nonnegotiable difference between them.
7.	Ernie Hayden et. al. 2014 [69]	Technical Report	Vulnerability management	Multi-Level	Defect management process with a framework is discussed.	Authors described the patch management as a solution for regular maintenance to increase the service life of software. According to it patch and vulnerability management remain one of the top requirements for a successful security program. This patch management is incorporated at design phase during software development process with a framework provided in the paper. It also includes defect management process. Its internal structure included 10 unique steps to be taken to search and patch defect and vulnerability flaws.
8.	Malik Hneif 2011 [70]	Research Report	Software durability and quality	Multi-level	Provided the guidelines to improving software durability.	Author has given the guidelines to improve quality in software nonfunctional attributes in his research. According to this research software development aims to produce software systems that satisfy two requirement categories: functionality and quality. One aspect of software quality is Nonfunctional Attributes (NFAs), such as security, performance, durability and availability. This research focuses on the point that software engineers can fulfill NFA requirements by using suitable guidelines during software development. The problem that causes complications in defining guidelines is the different effects of different guidelines on NFA quality and the relationships among the guidelines themselves. This research has given a step-

						by-step methodology that gives software engineers a suitable guideline set to improve the working life of software to get quality software. This paper studies the effects different guidelines would have after applying the guidelines provided by the author for improving life-span of software
9.	Van Der Linden 2010 [43]	Research report	Software durability	Single-level	Trustworthiness and human trust affect the service life of software and due to lack of durability, high failure rate is discussed.	Author has proposed a method for durability analysis of development systems in computing. To analyze software high failure rate, this report focused on a systemic view on software engineering. In this work durability is proposed as a key property of software development and also as a quantifiable sign of a system's ability to prevent bugs and failure. This research presented a view on software system development in computer science, and defined a method for maintainable development to measure durability of development systems.
10.	Basil Vandegriend 2006 [71]	Research Note	Challenges in maintenance	Single Level	Proposed idea for having two teams of maintenance to improve durability	Authors pointed about developing maintainable software. This study discusses about some guidelines on how to create software that is maintainable with minimum risk and impact. This paper discusses the specific challenges faced by developers during maintaining software. Author discusses that some organizations have separate software maintenance group from the software development group. Maintenance team may need to make emergency bug fixes or release defect fixes quickly, while at the same time development team keeps on working on new features.
11.	Robert C. Feenstra et al. 2009 [49]	Research report	Durability in software engineering	Single Level	Discusses the maintenance cost hike issues	Authors described the need and importance of durability in software engineering for different environment by reducing the cost of maintenance during early stages of development life cycle. They defined durable software as which doesn't changes with time. In addition, this study suggests a new reason why conventional hedonic methods may overstate the price decline of personal computers.
12.	Ruth Thomas 1994 [59]	Research report	Durable software	Single Level	Design should be focused to achieving the durability of software	Author described the importance of durable and low cost educational software. He also signified that there is need of durable and cheap software in the field of educational software. He has given a concept and issue to optimized software development life cycle for cost effectiveness. According to his research, developers should focus on the detailed design to achieve durability in s/w.

In table 2.3(a), most of the important contributions have been discussed. Lack of maintenance cost and time affect the durability of software, many contributors have addressed this problem. Further, some of the contributors have focused on trustworthiness, dependability, human trust and usability, separately, but no one discussed it all together. In addition, very few researches of the contributors have focused on software design to assure the durability of the software. Still, practitioners are trying to find the way to enhance the durability of the software. Without assessing security durability, there is no way to improve the life span of software. Some important work of security durability which is related to other specific areas is discussed in next section.

2.4 Literature Review on Security Durability

Researcher have found from the literature review of security durability that although there has been a little research work where security and durability have been addressed separately, but there has been no significant study that highlights the researcher's perspective of security and durability simultaneously. This presents the critical need of research on these two contrasting factors. Security and durability have been addressed independently by various foreign researchers in different fields. Some of the available literature has been shown in table 2.4 (a).

Table 2.4 (a): Pertinent Work on Security Durability

S. No.	Authors/ Years/ (References)	Contribution Type	Focus (Threats/ Goals/ Problems)	Assessment and Improvement (Single level/ Multi level)	Problem Addressed	Summary of the Contribution
1.	Alarifi A., et al. 2017 [46]	Model-driven	Security and Usability	Improvement at single level	Assessment data should be larger to apply results globally	Authors have proposed a structured inspection model for thoroughly evaluating the usability and security of internal and external e-banking assets. The authors have also demonstrated the insufficiency of existing security- usability models and have also applied their proposed framework to evaluate five major banks. The results clearly reflect several shortcomings regarding security and privacy features in banks.
2.	Nak Hee Seong et al. 2012 [72]	Research Paper	Security, Reliability and Durability	Multi-level	Real time case study might get more global results	Author discussed security and durability of phase change memory (PCM) allocation method. This study mentioned that a robust PCM design must take both security and durability issues into account simultaneously. By analyzing security durability prior in the design of PCM it can be secured from unknown dangers and malicious attacks. To improve durability and ensure security, the report describes a novel approach of security refresh. It also applied pinpoint attacks to understand the wear-out distribution using Security Refresh approach. The techniques proposed here will distribute the data

						placement more uniformly, improving its durability.
3.	Lori M. Kaufman 2009 [73]	Technical Paper	Security and durability in Cloud Computing perspective	Single level	Durability of security might have more focused in cloud computing.	Authors in "Data Security in the World of Cloud Computing" describes that the ability to employ scalable as well as durable, distributed computing environments within the confines of the Internet is known as cloud computing. The cloud computing world employs the virtual environment which further lets user's access the computing power which exceeds physical worlds. Author in this research explains as well as ways to improve the security of data in cloud computing environments and its durability.
4.	Troy Kinney 2002 [74]	Technical Paper	Durability assurance in hardware engineering	Single-level	Application of guidelines can be included.	Author from Catalytica Energy Systems Inc. (CESI) developed a novel catalytic combustion system (called XononO) that produces ultra-low emissions for natural gas fired turbine engines. In this project, the XononO system was installed on a Kawasaki M1A-13A 1.4 kW engine connected to the Silicon Valley Power electrical grid. The aim of this project was to demonstrate the ultra-low emission levels of the XononO system and to determine the system's Reliability, Availability, Maintainability and Durability (RAMD). The program did not get the quantification of maintainability and durability values because of the redesign of some key system components. This RAMD program also included guidelines to improve the reliability and durability of the whole system. Durability of the system mattered in the program as it was a real-world scenario of the software application

In table 2.4 (a), some important research has been discussed which are related to security as well as durability in various fields. In this row, software developers should focus on security and durability simultaneously during software development to improve the life span of security as well as software. Further, in **1992**, Parker D. B. said long security life span is needed to improve the user's satisfaction related to protecting user's data [40]. He also discussed about the challenges of high maintenance of security during use of software services. Due to high maintenance cost of security, practitioners are focusing on security design during a specified life span of software. According to Nathan Ensmenger, in the early 1960s, the development of the IBM OS/360 operating system has taken four years of maintenance time that absorbed more than 5,000 staff years of effort and cost the company more than half-a-billion dollars. This was making it the single biggest expenditure in IBM history [13].

To solve these types of issues, there is need to address the security durability during software development. Quantitative assessment is one of the most important methods to address, assess

and solve any issue. Security design during software development is a very crucial task. There are so many factors that affect the security and durability simultaneously including CIA. Every organization have its own methods and logics to develop the security as well as software design. All in all, this is a multiple decision analysis problem in perspective of durability of security, that's why researcher has taken a MCDA technique to assess the security durability. Literature of the assessment through MCDA has been discussed in next section.

2.5 Literature Review on Multi Criteria Decision Analysis

MCDM methods are the best techniques to solve the uncertainty problem of the choice of attributes to enhance the security of software. Fuzzy in hybrid with multi criteria approach has been used several times in the literature. Some of the pertinent works related to MCDA are discussed in table 2.5 (a).

Table 2.5 (a): Pertinent Work on Multi Criteria Decision Analysis

S. No.	Authors/ Years/ (References)	Contribution Type	Focus (Threats/ Goals/ Problems)	Assessment and Improvement (Single level/ Multi level)	Problem Addressed	Summary of the Contribution
1.	Kanza Gulzar et. al. 2017 [35]	Fuzzy Logic	Problem based on usability requirements	Multi-level	Only some factors of usability have been addressed. Other important factors are ignored	Authors proposed a fuzzy technique to prioritize usability requirements conflicts with respect to experimental evaluation. The report presented a novel framework that focused on the mapping of usability requirements attributes to the linguistic assessment from the users using fuzzy logic. Further, the proposed framework prioritizes conflicting usability requirements attributes. For implementation, the report has used MATLAB Fuzzy Logic Tool box. The proposed framework is aimed at helping the requirement analyst in taking better decisions by automating the whole process of identifying and resolving usability requirements conflicts.
2.	Afrin A. et al. 2017 [75]	Fuzzy AHP and Fuzzy TOPSIS	Problem based on Software Requirements	Single-level	Other techniques may be considered for requirements analysis other than group elicitation techniques	Authors presented a research on which they discussed about selection of software requirements using fuzzy AHP and fuzzy TOPSIS methodology. According to his research there are two types of software requirements functional and non-functional. There are number of factors effecting these requirements. In this condition group requirements elicitation technique is employed to select the useful requirements for developer. For eliminating the vagueness and selecting the best alternative for developers Fuzzy-AHP and Fuzzy TOPSIS

						methodology is used. The results of hybridization of MCDM methods have shown better results than considering only single methodology.
3.	Edmundas Kazimieras Zavadskas et al 2016 [57]	Hybrid Methods of MCDM	Problem based on sustainability of industries	Multi-level	Too many methods have been used and thus making it vague to come to a conclusion	In proposed methodology of hybrid MCDM methods to evaluate sustainability of industries and organization. This paper classifies sustainability problem into three domains based on economics, environmental and social aspects. Authors also found that which MCDM methods have been used the most frequently in developing hybrid approaches, we find that the most popular are the well-known methods that feature strong mathematical backgrounds and valuable characteristics, namely AHP, ANP, and DANP, TOPSIS, and VIKOR. At the end this work concludes that for better results it is advised to use hybrid methods in spite of a single MCDM method.
4.	Chong C. Y. et al. 2014 [76]	Fuzzy AHP method	Problem based on quality of virtual lab systems	Multi-level	Application of guidelines is not shown in the research	The authors raised the issue of Prioritizing and Fulfilling Quality Attributes For Virtual Lab Development through Application of Fuzzy Analytic Hierarchy Process and Software Development Guidelines. To lead to an effective virtual lab development, it is important to ensure that all quality attributes, stated in the service level agreement (SLA) are arranged in the priority wiser order. Priority assessment of the quality attributes is needed in order to focus on the higher priority ones while ensuring that the bare minimum expectation of the remaining ones is attainable. This paper uses a popular MCDM approach of FAHP for prioritizing quality attributes. After prioritizing quality attributes a set of guidelines is developed to achieve the quality attributes in virtual lab environment. The application of fuzzy AHP in the paper shown that participants ranked reliability, usability, efficiency, security, maintainability, and portability in decreasing order of priority, based on which a set of suitable, non-conflicting software development guidelines were determined.
5.	Davoud Goli 2013 [77]	Fuzzy-TOPSIS	Selection of Computer Security Software	Multi-level	Only a small number of software are used for research	Author presented an approach of Fuzzy TOPSIS methodology which selects the best computer security software among the seven established software. According to paper there are numerous antivirus software available in the market but there selection depends on some factors such as power, ease of use, cost etc. In this paper the most popular antivirus software in the market are short listed and studied to

						determine how they perform against the given criteria. Hence this problem is considered as MCDM problem and evaluated by the Fuzzy TOPSIS methodology followed by sensitivity analysis by considering three features as sensitive to results.
6.	Irfan Syamsuddin 2013 [36]	Ternary AHP methodology	Information Security Management	Multi-level	Other methods can be used to validate the results	Author has presented a research article that discussed about the decision analysis based on Ternary Analytic Hierarchy Process (T-AHP) as a novel model to aid managers who were responsible in making strategic evaluation related to information security issues. Sensitivity analysis was also applied to extend this analysis by using several "what-if" scenarios in order to measure the consistency of the final evaluation.
7.	S. K. Dubey et al. 2013 [78]	Fuzzy Multi Criteria weighted average method	Software Quality and usability	Multi-level	Sensitivity analysis is not evaluated to show the variations in results	Authors proposed a methodology for quantifying the usability rating of software using a fuzzy multi-criterion weighted average approach. Usability is an important attribute of software quality. A case study of MS Word 2003 was then taken to validate the feasibility of this approach. Attributes of evaluation has been taken from ISO 9126-1. After it hierarchy of attributes has been designed to analyze the usability of case study. Rating of attributes is evaluated in this paper to calculate the final usability of Ms Word 2003. This usability model can be used by other developers to evaluate usability of any software.
8.	Li Shi et al. 2012 [79]	TOPSIS method and entropy method	Software trustworthiness	Multi-level	Real time case study is shown with validation also	Authors have presented a research article that has proposed a new software trustworthiness evaluation approach based on combination weights and improved TOPSIS methods. In this work entropy weighting method is used to calculate objective weights. Then the method uses the combination weighting method to calculate the priorities of the attributes of trustworthiness. Further this data is used in TOPSIS method to evaluate the trustworthiness of PLM software, which is taken as a case study of an aircraft equipment manufacturer in china. Sensitivity of analysis is done on the case study to validate the results
9.	Nadir Omer FadlElssied 2011 [80]	Fuzzy set theory	Software Security	Single-level	There is need to find more efficient approaches to improve the security	A research article was presented which discussed about the fuzzy set theory is very useful for evaluation of e-government security. His paper has investigated and reviewed the application of Fuzzy algorithms in the field of e-government security. A comparison between five approaches based on Fuzzy has been described in his article. In the

						end authors concluded that there is need for new evaluation methods in artificial intelligence arena to obtain the good performance particularly in the evaluation of effectiveness and efficiency. It also argues that present approaches present good results but still fully secure software is out of reach i.e, there is lot to do in this field.
10.	Aşkın Özdağoğlu et al 2007 [81]	AHP and Fuzzy AHP	Case study of employee selection	Multi-level	Fuzzy-AHP is best suited for decision making problems	In this research the author compares the methods of AHP and Fuzzy-AHP using a case study. The case study adopted here was about employee selection for shop floor of manufacturing platform applied in a company from food industry. In order to avoid the risks on performance, the fuzzy AHP, a fuzzy extension of AHP, was developed to solve the hierarchical fuzzy problems. According to the research Fuzzy AHP is best suited for the decision making in the applications where data is to be retrieved in linguistic values. Fuzzy logic method is capable of handling ambiguousness in the linguistic data and best suited for the applications in software industry. AHP uses linguistic values and evaluates priorities by using a weighting process within the current alternatives by pair wise comparisons
11.	Liming Zhu et al. 2005 [37]	AHP	Software Architecture Evaluation	Multi-level	Software architecture evaluation using analytic hierarchy process.	Authors presented a research paper on Tradeoff and Sensitivity Analysis in Software Architecture Evaluation Using Analytic Hierarchy Process. Software architecture evaluation involves evaluating different architecture design alternatives against multiple quality-attributes. These attributes typically have intrinsic conflicts and must be considered simultaneously in order to reach a final design decision. In this paper, authors propose several in-depth analysis techniques applicable to AHP to identify critical tradeoffs and sensitive points in the decision process. Also, they apply their method to an example of a real-world distributed architecture presented in the literature. The results were promising in that they make important decision consequences explicit in terms of key design tradeoffs and the architecture's capability to handle future quality attribute changes.
12.	Van Laarhoven et. al. 1983 [82]	Fuzzy-AHP methodology	First use of Fuzzy AHP	Multi-level	Proposed the Fuzzy AHP for implementation with triangular fuzzy numbers.	Author proposed the first method of implementing Fuzzy AHP in 1983 in which triangular fuzzy numbers were compared according to their membership functions. Fuzzy AHP method was used in various research areas for decision-making in different fields such as selecting, prioritizing, and evaluating so on. The Authors applied the method at

						two distinct levels. The first level was to find fuzzy weights for the decision criteria, and second was to find fuzzy weights for the alternatives under each of the decision criteria. Then by an appropriate combination of these results, fuzzy scores for the alternatives were achieved. According to this research these fuzzy scores can be used by the decision-makers to make a choice for one of the alternatives.

In table 2.5 (a), most of the important literature review has been done. It can be seen that MCDA is very popular technique for assessment of the software development issues. Further, it is not only helpful technique to assess the security durability but also, results from this technique is helpful in real scenarios as seen in the literature. Hence, MCDA technique is used by the researcher in this work and implementation process is discussed in next chapter.

2.6 Expert's Saying

User's needs are changed related to security in these days because new threats are generating the challenges of maintenance process. Further, users do not want the interruption in the business for long time through the maintenance process of security and software services. That's why, practitioners are continuously trying to solve this problem. In this regard, many of the practitioners have focused and put forward their comments related to these issues. *DeMarco and Lister* quoted "Quality is free, but only to those who are willing to pay heavily for it." [83]. Standards and meanings for a quality product change with the time. At first it was fulfilling the user's basic needs. Now security needs are added to the basic requirement in software.

The security community has recognized the need for incorporating a human dimension into its work since a very long time. Surprisingly, however, understanding user needs for security is increasingly appearing these days. The importance of security has been recognized in recent years, may be due to a number of increased cases of attacks reported by the media [26]. *Lehman's* first law of maintainability states maintainability as "A program that is used undergoes continual change or becomes progressively less useful" [84]. Strengthening Lahman's fact *Nathan Ensmenger* says "Maintenance is a Misnomer" and also a "Dull and Dirty work of Maintenance" [13]. In this row, *Mr. Tekinaslan* state that the first 90 percent of the code accounts for the first 90 percent of the development time and the remaining 10 percent of the code accounts for the other 90 percent of the development time [85]. Regarding Rapid Development, *Steve McConnell* does an interesting experiment by giving the same piece of

software to develop to five different development teams. Each development team had a different list of objectives which contains memory use, output readability, program readability, minimum statements, and minimum programming time [86]. As a result, only four among five developers were able to achieve at least two objectives. This shows that achieving all the objectives in one development is not possible.

Robert C. Martin states that "It is not enough for code to work." in Clean Code: A Handbook of Agile Software Craftsmanship [87]. Durable and secure software development aims to develop the software whose security durability can be quantified. *Kevin Mitnick* says about security that "Companies spend millions of dollars on firewalls, encryption and secure access devices, and its money wasted, because none of these measures address the weakest link in the security chain" [88]. Weakest link in security can be identified by finding vulnerable holes in software design. According to *Bruce Schneier* "If you think technology can solve your security problems, then you don't understand the problems and you don't understand the technology" [89]. Technology doesn't mean that it will secure itself. Security is not built in a day, it is developed in steps, security by design, security by development etc. *Gary McGraw* states that software security is about integrating security practices into the way you build software, not integrating security features into your code [4]. Hence, designing security through the steps increases security durability of whole software life span. Although, *Thomas C. Gale* observes the design of security as "Good design adds value faster than it adds cost" [90]. Further, *Gabriel Morgan* of Microsoft Corporation said that "Build high-quality software, leverage industry practices, and plan to build quality into your solution; but be sure to prioritize carefully" [91]. In 1995, *Sutherland* described the cost of maintenance during use of software services in the United States, which has been estimated more than $70 billion annually for more than ten billion lines of existing code [92].

According to these statements through the practitioners, it is clear that there are so many loop holes in software design and to improve the life span of software services, security is key point. Without considering security durability, there is no way to improve life span of software services. From the pertinent review of literature, researcher found some major findings which are discussed in next section.

2.7 Major Findings from the Literature Review

There are a number of existing models that incorporate maintenance in the SDLC [26]. But still maintenance cost and time invariably increases as the software services and its security issues increases. Also, there is no focus on increasing life of security and decreasing time and cost given to maintenance. Security durability assessment describes about the same problem and its solution. After the thorough literature review of the related fields, it is found that MCDM methodology (Specifically Fuzzy AHP) not only improved the accuracy of security assessment, but also is more objective at the same time. These statements are even more true in the case of software security because the growth of security is still in its infancy and there are very limited established references [39]. The main aim of this work is to address the security durability that can provide a solution with higher security for software services that may be enhanced through the assessment. Assessment of security durability is another approach to reaching a high level of security life span. Therefore, the approach of this report is treating various issues including durability, optimal security maintenance, reducing cost and time to maintain security for longer use. After a careful and centric study of some available approaches of the software security, software durability, security durability and MCDA, the following inferences are drawn:

- After deployment process, time and cost for security maintenance is increasing. To get rid of this increasing maintainability time and cost, practitioners are continuously trying to integrate new features into security.
- Users want the software services for longer duration with minimum efforts to maintain its security. So, that continuity in their work is not disturbed, again and again.
- Users want software which can maintain security itself for a long-time duration.
- Users want secure and durable software, so that their business is not interrupted through security failure. And unfortunately, they are not still satisfied.
- In the field of software development, every organization follows its own mechanism for securing the software. There is no common mechanism and framework for it.
- There is need to collect expert's advice in this field and develop a framework which provides longer security software.

- From the literature review, it is clear that MCDM approaches are one the most important techniques which is useful to quantitatively assess the software security measures.
- There is no single work available to predict security of software for a better life span of software services. There is need to evaluate security durability at early stage of development process to minimize the maintenance efforts.
- To assess and improve the security durability, it is needed to bridge the gap between security and durability through their attributes.
- For measuring and improving security durability, current measures depend greatly on threat models and attack types. They offer little information when the development environment changes.
- MCDA approaches may provide better results for security durability assessment and improvement.

2.8 Conclusion

From the literature review of previous works and best practices, researcher found that despite the necessity of establishing security and durability simultaneously during software development, especially in the design phase, there has been a gap between attributes of quality. To fill the gap, the relationship between security and durability should be established. The available literature can be divided into three patterns: in the first category, there are the approaches which try to improve security during the development life cycle. The second category of approaches, tries to find the methodologies which improve durability of the software, either after development or at later phases of software development. The approaches in the third category improve security through MCDM techniques. The identified approaches in the third category are important and can be used to estimate and enhance the security durability of software service life span. Unfortunately, no work is identified to estimate or predict security durability at early stage of development life cycle. Hence, there is an urgent need to develop a security durability centric approach to assess and improve the security durability.

CHAPTER - III
SOFTWARE SECURITY DURABILITY ASSESSMENT

3.1 Introduction

The development environment of software in the early twenty-first century creates new challenges for all, including the security developers [13]. Earlier practices have shown that the security of software is not that high as it could be. Development organizations spend a comparable big sum of money and effort on resolving security issues during the previous stage of secure software development [48]. Though they do not worry about the longevity of the security offered. The process of integrating security in the software during its development is called secure software development life cycle. With the increasing demand of secure software, developers are facing new challenges to fulfill customer's requirements while developing software [146]. In the security perspective, software development includes security attributes, security strategy, security design, security testing, and security management. Unfortunately, security is often integrated only in isolation and late in the process. Organizations impose development constraints due to cost, time-to-market requirements, productivity impact, customer satisfaction concerns, etc. This gives results with improper development of secure software and less durable security [46].

That’s why, one of the issues of software that has received significant attention in recent years is durability [93]. Durability and security are high in demand in present era [147]. This demand of security necessity is making companies successful or unsuccessful in the market. While there has been an agreement between the industry persons and researchers, security durability is far away to be measured. Further, durability is defined as the longevity or timeliness of security for a specific duration. Security of the software starts to fail after a period of time. This expiration time of security can be affected by the durability. Software security with poor durability is likely to fail in a highly competitive market; therefore, software developing organizations are paying more attention towards ensuring the durability of their security. To develop cost-effective security durability, there is a need to investigate the connection between durability, its characteristics, and security [148]. To reset the evaluation, it is important for security experts to review the issues of software development, security design and user’s satisfactions. The relationship between software issues, user’s needs and evaluation is shown in figure 3.1 (a).

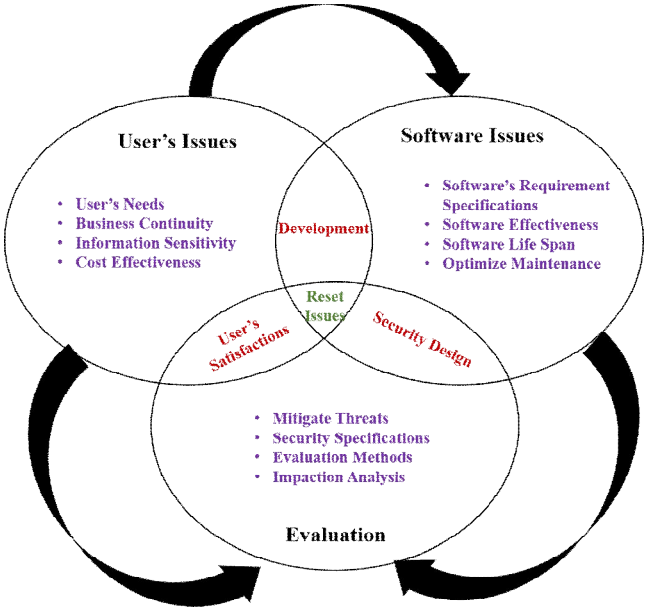

Figure 3.1 (a): Relationship between Software, Security, and User's Needs

In figure 3.1 (a), three steps are very important to reset the issues. User's issues and software issues are main for development, whereas user's issues and security evaluation are the key behind user's satisfaction. Also, software issues and security evaluation are the basis for security design. Further NASA presented a report on expenses on software maintenance. This report described that software maintenance has invariably increased [50]. For reducing these expenses, there is a need to develop flexible, durable and secure software. According to a report in 2016 [51], it was found that 60% of time and cost are being consumed on security maintenance. These multiple reports on software maintenance focus on a single issue of non-durable software security.

According to another report, the service life of working software affects durability during the former stage of software development [52]. This report discussed durability challenges that it depends on the dependability and trustworthiness of developed software and also discussed differences between durability, consistency and survivability of software. For overcoming this problem and for focusing on the durable security to enhance the working life of software, our contribution proposes durability as a security factor. It has also been found that if it is possible to judge the working life of the secure software, the cost and time incurred for maintenance can be lessened [148-150]. During development life cycle, identification of security attributes may optimize the maintenance issues and thus decrease time and cost incurred on it [58]. Nowadays, users and organizations depend on technology and technology is nothing without software

applications. Developing secure software is a complex concern and security attributes must be considered as important tools of security during software development. Identification of new security factors helps to improve security during software development [18].

3.2 Proposed Definition of Software Security Durability

The importance of software in our lives is growing daily. People's personal and professional lives can greatly be enhanced by the presence of highly secure and durable software and can greatly be imposed upon by the presence of poor quality software. Most complex software systems, such as airplane flight control or nuclear power plants, depend critically upon the durability of their secure software. In today's world, organizations are busy in understanding and mitigating security challenges during the software development life cycle. There are some key characteristics of security, focusing on which may help to address these challenges directly or indirectly. One of these characteristics is durability. It may also be called as working life or longevity of security [18]. The security durability of software is highly essential in sensitive fields including banking sectors etc. [13]. Security is directly involved in service life of software. Durability is further directly or indirectly involved in the security of software and vice-versa. Through the literature review of previous work and best practices, the researcher has defined the **security durability/durable security** as [110]:

> ***The ability of software to secure itself for the expected life-span***
>
> **or**
>
> ***The ability of software to withstand attacks for the expected life-span***

Durability means how long a software security solution will function effectively and meet the security requirements. There are several reasons for organizations to integrate durable security during software development as:

- To provide longer security in the given service environment, thereby mitigating security challenges [94].
- To reduce maintenance time by reducing the effort needed to fix bugs by delivering durable and secure software [29].

These are two main reasons to examine the security and durability simultaneously for addressing, assessing and improving the security durability. There are so many attributes of

security and durability which are related to each other. These attributes are useful in assessing the security durability. Further, next section is discussing about identification and mapping of these attributes.

3.3 Identification of Software Security Attributes

Software security is implemented to have the best practices to enhance technical methodologies and consider security assessment at the very beginning of the software life cycle. These best practices help in identifying and comprehending the normal threat, design for a secure lifetime [101, 150-152]. In addition, software security is about building secure software which means making software with a secure design and also ensuring that every developer who is involved in the process understands the importance of security [4]. The science of software security concerns constructing secure software. An effort has been made to enhance security durability by reducing the maintenance time and cost by analyzing the whole software development process using durability as a key factor. Software security depends on its attributes for estimation. Hence, for estimating security durability we need to first encounter the security attributes. Some of the security attributes with their definition are given in table 3.3(a)

Table 3.3(a): Software Security Attributes

S. No	Security Attributes	Definition	References
1.	**Confidentiality**	In the context of security, confidentiality refers to the allowance of authorized access to sensitive and secure data.	[16]
2.	**Consumer Integrity**	Consumer integrity is defined as the attribute maintaining the consistency, accuracy and trustworthiness of consumer all over the life cycle of a software product or security.	[17]
3.	**Authentication**	Authentication is the factor which is responsible of the identity of the user profile. It is the process of determining whether a user is, in fact who it is declared to be.	[16]
4.	**Reliability**	Reliability is the ability of security to consistently perform according to its specifications. It is considered to be very important aspects while designing security.	[9]
5.	**Maintainability**	It is the parameter concerned with how the system in use can be restored after a failure. Also, it is the probability that a secure system can be repaired in said environment or situations.	[41]
6.	**Accountability**	This term means that every individual user who works with secure system should have specific responsibilities for security assurance. These tasks include individual responsibilities that are part of the overall security plan and can be vulnerable by responsible person such as a developer.	[12]
7.	**Survivability**	Survivability is the ability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents.	[23]
8.	**Availability**	Availability means that information is accessible by only	[22]

		authorized users. Availability, in the context of a computer system, refers to the ability of a user to access information or resources for a specified duration.	

In table 3.3(a), the most important attributes that affect the life span of security of software services are described precisely. Further, some attributes of durability that may affect the security of software are discussed in next section.

3.4 Identification of Software Durability Attributes

Security is a noteworthy factor of software quality that affects its factors such as usability, maintainability, etc. [12-13]. In the earlier years, security needs were identified and categorized in the form of security attributes. These attributes are used to enhance security of software through identification, classifications and measurements. These attributes depend on other quality attributes for deciding how much software is secure; same as the durability decides for how longer software will be secured.

Further, it is difficult to assess whether the software will be durable for a sufficient time period to handle unpredictable future needs. Durability concept is multi dimensional and multi criteria in nature. By software designer's, most of the time is consumed on designing high security of software that meets the desirable standards of software quality. To assess software durability, its attributes should be identified in a quantified manner. The attributes are same as the needs for quality software that meet durability also [1-2]. Researcher found most important durability attributes including dependability, human trust, trustworthiness, and usability that are discussed in table 3.4(a)

Table 3.4(a): Software Durability Attributes

S. No.	Durability Attributes	Definition	References
1.	**Dependability**	Dependability refers to the ability to deliver service that can justifiably be trusted.	[12]
2.	**Trustworthiness**	Trustworthiness is assurance that the software will perform as expected.	[44]
3.	**Human Trust**	Human trust is willingness to rely on the software with confidence.	[38]
4.	**Usability**	Usability refers to how well software can be used by a particular user to reach quantified results with effectiveness and satisfaction.	[46]

Table 3.4(a) is showing the definition of the software and security durability attributes that affect the life span of software. Among these software durability attributes researcher has recognized three attributes which also affect security durability i.e.; dependability,

trustworthiness and human trust. The description and relationships of security and durability attributes are specified in the next section.

3.5 Relationship between Security and Durability Attributes

Durability is said to be what occurs when security of software goes out all over. As time passes, use of software, need for updating security increases because new security threats are generated day by day [48]. If these threats get active then security will fail and as a result software will crash. Through the literature review of previous work, the researcher has identified three relevant security durability attributes [12] including dependability, trustworthiness and human trust that is to be used for measurement of security durability. Each of them is described in the following section with their relation to the security.

3.5.1 Dependability

A computer is called secure if the user can depend on it and its software to work as expected [59]. This definition is controversial as it implies that security exists in the user's expectations of computer and software behavior. It is useful, however, in underlining the importance of dependability in computer security. Security durability is affected as well by dependability and its other co-attributes. Security durability implies that security which works for long duration. While according to dependability definition, it is inferred that user's expectation of secure software service life span is important. Hence, in these terms dependable software or security helps building durability of security stronger. The definition of dependability is given in table 3.4(a). This definition stresses the need for justification of the trust. Hence, it is directly related to security attributes such as confidentiality, authentication and reliability. The alternate, quantitative, definition that provides the criteria for deciding if the service is dependable is: dependability of the software is the ability to avoid service failures (including security service failures) that are more frequent and more severe than is acceptable to the user(s). The quantitative definition formulates that dependability is also related to availability and maintainability.

3.5.2 Trustworthiness

The trustworthiness of the secure software is that it performs as intended for a specific purpose, when needed, with new changes that have been done in recent, and without unwanted side-effects, behaviors, or exploitable vulnerabilities. The definition of trustworthiness is given in

table 3.4(a). Hence, according to its definition, trustworthiness depends on availability, reliability, maintainability, accountability and survivability. Further, security durability works for a specified time period with strengthening the maintainability of security of software services, henceforth improving the trustworthiness of security. The term operational resilience which strengthens trustworthiness of security is a set of techniques that allow people, processes and informational systems adapt to changing patterns [65,113]. This term directly points to the maintainability that affects the security life span of software. The quantitative definition formulates that trustworthiness is also related to availability, reliability, accountability and survivability.

3.5.3 Human Trust

In relation to human–human interaction, human trust is mostly defined as a sensitive issue where the trusted party has a moral responsibility to the trusting party. In software terms, Consumer's trust on the developers is identified as human trust. Consumer when uses secure software, their trust is dependent on security design of the software that it will work for an expected duration and secure their data or information. Security durability and human trust are the attributes that strengthen each other. A long life service of security will perform according to the human trust and improve the consumer reliability on organization's software services. The definition of human trust is given in table 3.4(a). According to its definition, it is found that five security attributes that may affect the human trust includes reliability, consumer integrity, accountability, confidentiality and authentication. Human trust is invariably dependent on these factors. Dependency means strength of these five factors help in building stronger human trust.

3.5.4 Other Relating Attributes

A considerable measure of research is accessible, endeavoring to comprehend and characterize the manners by which the security of software can be upgraded [1-2]. While there has consistently been a hole among hypothesis and practice which is difficult to fill completely, the issue can be diminished by building up a common terminology and enhancing the availability of research results. With the investigation of security and durability in this work, it has been attempted to create hypothetical research for evaluating security durability. Durability is a vital attribute to provide security of software for longer period. To assess security durability during software development, identification of security as well as durability attributes is essential [4,

43]. Therefore, developers need to understand how to relate security attributes with durability attributes and measure the impact of these attributes on security life span of software. Mapping of security durability attributes with security attributes is necessary to ensure it [33]. Security durability assessment outcomes may allow decision makers to make appropriate decisions as well as action [95]. However, to be able to take appropriate action, decision makers are not only needed to know about security and durability attributes but their mapping also and mapping of security durability attributes is shown in figure 3.5.4 (a).

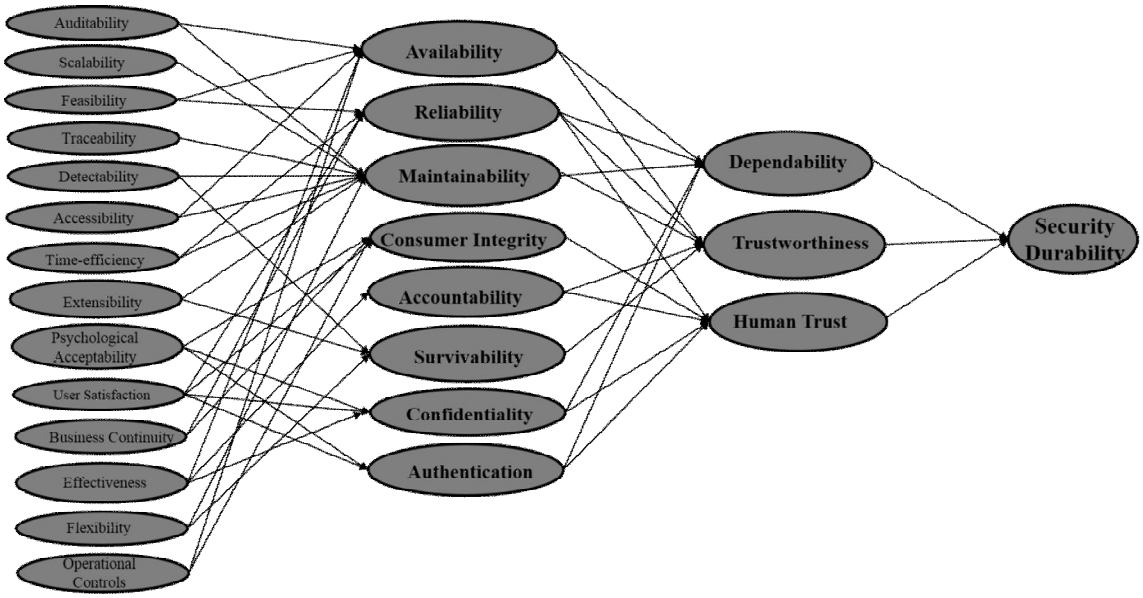

Figure 3.5.4 (a): Affiliation between Security Factors and Key Determinants of Durability

In figure 3.5.4 (a) many attributes of security affect durability as confidentiality affects software effectiveness evaluation, user satisfaction and operational controls; availability affects auditability, feasibility, accessibility, software effectiveness evaluation, operational controls; reliability affects feasibility, time-efficiency, user satisfaction, business continuity; maintainability affects auditability, scalability, traceability, detectability, accessibility, time-efficiency, extensibility, effectiveness, flexibility; consumer integrity affects psychological acceptability, user satisfaction, business continuity; accountability affects software effectiveness evaluation; survivability affects detectability, extensibility, flexibility; authentication affects user satisfaction, psychological acceptability, software effectiveness evaluation and operational controls. Level wise full descriptions of the above hierarchy or mapping are given in the next section.

Security Durability relating with Confidentiality

Confidentiality discusses of securing data, disclosure of which may harm the information [58]. Some developed software have very specific policies regarding access to, and release of confidential data. Security developers are less likely to have fixed policies concerning appropriate transformation of information between users [51]. In general, confidential data for specific time duration is the boost for the security durability of software services. Confidentiality and security durability are related to each other. These two relating concepts (confidentiality and durability) need to be notable. With durability, confidential information of data will be more secure for a specific period. Confidentiality, supported by durability, is primarily the developer's duty during early stage of secure software development and that is not to disclose his user's confidences. Confidentiality is further affected by other security sub-attributes including accessibility and user satisfaction which in turn affects dependability and human trust.

Confidentiality is equivalent to privacy [52]. It is defined as a factor that is responsible for preventing sensitive information from reaching the wrong people, while making sure that the right people can in fact get it. On the other hand user satisfaction is a degree of how secure services are being provided by an organization to meet customer expectation. User satisfaction is responsible for ensuring that the client is satisfied with the privacy settings of security and gets a trustworthy security of software. Software effectiveness evaluation is the degree of evaluating that software is successful in producing a desired result. It is ensured by confidentiality that a software security is effective to what extent. Operational Controls are safeguards and countermeasures to avoid letting security operate properly [40]. Also, operational controls make security controls and policies to maintain privacy for users for a time period and ensure confidentiality of security.

Security Durability relating with Consumer Integrity

Consumer **i**ntegrity implies keeping up and guaranteeing the exactness and consistency of data of information [23]. Further, it is a quality of appeal established by the ethical assurance and resolution essential to continue dependability between the four components of consumer integrity including psychological acceptability, user satisfaction, business continuity and scalability. Consumer integrity is violated when information is actively modified in transferring the information. This implies that data cannot be modified in an unauthorized or undetected

manner. Lack of consumer integrity is one of the most important reasons of security failures. Consumer integrity and durability cover a gap in knowledge on security assessment required for delivering secure software. Durability ensures that the consumer integrity of the security of software is maintained for time duration. For measuring and assessing security durability, consumer integrity plays an important role.

Consumer integrity is maintaining the accuracy of the data and trustworthiness of consumer related to security [12]. It has four components to consider viz psychological acceptability, user satisfaction, business continuity and operational controls. Psychological acceptability is stated as the acceptance in human psychology which is a person's assent to the reality of a situation, recognizing a process or condition without attempting to vary it, or protest against it. Consumer integrity is about maintaining the integrity of user and getting his assent that there will be no change in the integrity during the working life of security. User satisfaction is also important while working on consumer integrity, as the integrity can be maintained after the satisfactory responses from the users. Business continuity encompasses a loosely defined set of planning, preparatory and related activities for software security which are intended to ensure that an organization's critical business functions will continue to operate within a period. To uphold the trust of consumer and maintain the integrity, business continuity of security should be held. Operational controls are safeguards to avoid letting security run properly. Further, it helps in maintaining the integrity of consumer by properly running the security even in opposite situations.

Security Durability relating with Authentication

Authentication accepts on the way that just those clients who benefit to be credible and checked will be getting satisfactory responses [12]. In security terms, authentication is the process of attempting to verify the digital identity of the sender of a communication. It is often controlled at the interface stage. Authentication is well affected by the user satisfaction and software effective evaluation. Software effective evaluation is to determine how well the software meets the needs of secure software development. Authentication is effected by dependability and human trust, which in turn affects durability. Ensuring and strengthening authentication in secure software development is important as it further enhances the security durability. Therefore, while measuring security durability; authentication is a key factor is to be considered.

Authentication is the process of determining whether someone or something is in fact who or what it is declared to be. It is composed of four factors including psychological acceptability, user satisfaction [135], software effective evaluation and operational controls. Psychological acceptability is already defined above. Authentication works on the psychological tendency of the user's to accept the changes that have been made to the security within a time period. User satisfaction is a degree of how secure services provided by an organization meet customer expectation. User satisfaction is responsible for ensuring that the customer is satisfied with the authentication settings of security and have gets an authenticity while accessing security of software. Software effectiveness evaluation is the degree of evaluating that software is successful in producing a desired result. It is ensured through authentication that a software security is how much authentic and maintaining the privacy.

Security Durability relating with Availability

Availability certifies that the consistent information about data and services will be available, when it is demanded for duration of time [12]. Availability affects the degree to which various kinds of users can depend on such as dependability and trustworthiness of software. If there is improvement in the availability, then security durability also works with less complexity as well as increases the security. Being trustworthy and dependable is such an important thing to improve the quality of secure software that it makes durability as a strong characteristic for consideration. While users are likely to allow interruption from their security services of software, they demand a much secure guarantee that their security of software is never permanently lost due to failure. Security durability thus increases through the availability of security. Availability of security increases the capacity of security durability through assessing the time duration of security.

Further, availability has five components that are auditability, feasibility, accessibility, software effective evaluation and operational controls. Availability is defined as keeping data available to authorized users only, while auditability records the facts such as time and date of audit occurrence. Availability goals are achieved by achieving auditability in security. Feasibility specifies the requirement to be fulfilled before the development of security status. Availability makes the security durable by providing the feasible needs of user for a long time run. Accessibility is the ability of a user to access the contents of secure data with authorization and availability. A secure, durable and accessible security makes availability of data to the authorized user. Software effectiveness evaluation is the measurement of effectiveness of

security settings applied to software. To ensure availability of security one must ensure the software effectiveness evaluation. Operational Controls are safeguards and countermeasures to avoid letting security operate properly. Also, it makes security controls and policies available to user for a time period and ensures availability of security.

Security Durability relating with Survivability

Survivability is the ability of software to achieve its assignment in a timely manner, accidents, despite attacks or failures [12]. It ensures that software is providing vital properties to users even in the occurrence of an attack, software cannot have failure arguments. Further, it can be ensured for a time-period than it will be easy to manipulate the security according to developer's choice. Hence, survivability should be durable so that security of software is durable as well as survivable. As an idea and as a preparation, survivability is engaged with security durability in this research. Understanding the significance of survivability, it contributes to lasting development and assignment completion. Survivable security necessity continues to be delivered in unkindness of either malicious or accidental harm. When developers tend to improve durable security, within this process survivability will also be improved. Timeliness, flexibility, user satisfaction are the main attributes which affect survivability and ensure that security is trustworthy and dependable.

Survivability is the ability of a security system to remain working even in the opposite situations. It has three components that are detectability, extensibility and flexibility. Survivability depends on detectability to detect the failure occurred during the working life of security. Detectability is an attribute that is responsible for detection of security failures or crashes in software in duration of time. The ease with which security can be enhanced in the future to meet changing security requirements or goals is called extensibility. Security can be extensible when it is survivable in the changing environments. The capability of secure software to respond to potential internal or external changes affecting its value within timely and cost-effective manner is stated as its flexibility. The survivable security of software is flexible to the changes made to it.

Security Durability relating with Reliability

Reliability is one of the peak significant quality characteristics of safety-critical software. Reliability discusses to the dynamic performance of the secure software. It is the degree to which a work product operates without failure under given conditions during a given time

period. Further, it means to prevent security from failures and to make strategies to maintain its trustworthiness. So, it can be estimated that durability enhances security with the help of reliability as well as human trust. Being a reliable, secure software means to maintain feasibility and trust of users. Trustworthy software is reliable and feasible within a time duration that is if an error occurs, it deals with it, without any specific change in secure software. There is very little difference between durability and reliability as the durability of a product or service implies, it is reliable - will perform as expected, for the duration expected. Durability is "the ability of an item to perform its required function under stated conditions of preventative or corrective maintenance until a limiting state is achieved" and on other hand reliability is defined as "the ability of an item to perform a required function under a stated period of time".

Reliability has four components that are feasibility, time-efficiency, user satisfaction and business continuity. Definition of reliability states that the security will perform as it is intended with particular specifications and within a particular time-period while the feasibility study specifies the requirements to make a system reliable for a time period. As per the definition of reliability, it is to be stated that reliable software and its security is time-efficient also. Time-efficiency states the ability to deliver the appropriate performance of securities within a time, a time period and under stated conditions. User satisfaction focuses on satisfying the needs of the users as per the feasibility analysis and provides a reliable security of software to the users. Hence, reliability depends on user satisfaction to fulfill its compelling needs through security durability of software. Also, it ensures that business continuity of the company is fulfilled. Business continuity is focusing on ensuring that organization's business functions may operate within a time period and under stated conditions.

Security Durability relating with Accountability

Accountability is an essential security concept. It means that every individual who works with secure applications should have specific responsibilities for information assurance that it should not be easily accessible for a time period [12, 96]. This specific time period implies to state durability importance in achieving accountability. Software which is fully secure is distinguishable as well as accountable and security with enhanced durability contains accountability as well. Accountability in security means that every individual who works with the security as well as software should have specific responsibilities for assurance. Software effectiveness evaluation means the evaluation of effectiveness of software when the security

settings are being applied to it. To ensure accountability of security one should measure the effectiveness of software after the accountability settings are being drawn upon it.

Security Durability relating with Maintainability

Maintainability is very important factor that affects life span of security when there is any modification done in software [12, 42]. These modifications could influence mechanisms, features, and interfaces when changing the software's functionality in direction to meet new business needs as well as security needs. Durability of security ensures that there is no major need of maintenance of security for a specific duration and security works properly for that duration. Maintainability is a factor by which defaults are corrected after the delivery of software.

According to the hierarchy, maintainability has eight components including auditability, scalability, traceability, detectability, extensibility, flexibility, accessibility and time efficiency. An audit is effectively processed when the auditors have full access to security process and documents for managing the security related issues timely. While maintainability is the ease with which software or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment. Auditability helps in improving maintainability of security by providing practitioners the understanding of the past and present documentation during early stage of software development. Scalability is the ability of the software that operates in conditions when it is changed in size or volume according to the user's needs. Adaptive maintenance decides on scalability; hence maintainability and scalability may be related.

Traceability is the ability to maintain the history and records of previous data of security design even if the changes occur. One should ensure traceability of the security when accessing the maintainability of the software when software is in use. Detectability is the ability to detect the failures or crashes of security during use of software for a time period. Maintainability composes of these attributes to trace the error and faults during the working life of secure software. Extensibility is defined as the ease with which security can be enhanced by making changes in security needs. Maintainability also depends on extensibility to improve the working life of software so that the cost and time incurred in security maintenance can be lessened.

Flexibility is defined as the capability of secure software to respond to potential internal or external changes affecting its value within timely and cost-effective manner. Hence, it is clear

that maintainability depends on this factor to conform to changes implied on security. After maintenance process is over the security of software should be accessible to the user here accessibility factor of security come into existence within maintainability. This states that accessibility is the degree to which the security services or environment is available to as many people as possible. Time efficiency is stated as the capability to provide appropriate performance of a security, relative to the amount of resources used understated conditions within specific time duration. It ensures that the security that is to be maintained is available to the user under said conditions and time period.

Security is one of the most significant quality factors that are concerned to software developers as well as consumers. As new security needs arises, security attributes must be identified and their relationship must be composed for their contribution during software development. Durability of security cannot be achieved without considering the important attributes. The connection between security and durability is a significant issue to improve the security for a life span. The sub attributes play very noteworthy role in the assessment of security durability. These sub attributes affect security attributes indirectly or directly. The categorization of sub attributes helps in better assessment of the security durability during software development. Further, the definitions of these sub attributes are shown in Table 3.5.4 (a).

Table 3.5.4 (a): Definitions of Durability Sub Attributes

S. No.	Sub Attributes	Definitions	References
1	**Auditability**	The capability of supporting a systematic and independent security process for obtaining audit evidence and evaluating it accurately to determine the extent to which audit criteria are fulfilled.	[12, 40]
2	**Scalability**	Scalability is the measure of how well a security can grow to meet increasing performance demands.	[12, 42]
3	**Feasibility**	A feasibility study is an analysis of how successful a project can be completed, accounting for factors that affect it such as economic, technological, legal and scheduling factors.	[12, 33]
4	**Traceability**	The ability to trace security representation, or actual program module back to requirements for security.	[12, 47]
5	**Detectability**	Detectability is an attribute that is responsible for detection of security failures or crashes in software in duration of time.	[12, 29]
6	**Accessibility**	Accessibility is the degree to which a software security service, or environment is available to as many people as possible.	[12, 18]
7	**Time-efficiency**	The capability to provide appropriate performance of a security, relative to the amount of resources used understated conditions within specific time duration.	[12, 24]

8	**Extensibility**	The ease with which security can be enhanced in the future to meet changing security requirements or goals.	[12, 21]
9	**Psychological Acceptability**	Acceptance in human psychology is a person's assent to the reality of a situation, recognizing a process or condition without attempting to change it, protest.	[12, 15]
10	**User satisfaction**	User satisfaction is a degree of how secure services provided by an organization to meet customer expectation.	[12, 11]
11	**Business Continuity**	Business continuity encompasses a loosely defined set of planning, preparation and related activities for software security which are intended to ensure that an organization's critical business functions will either continue to operate within a period.	[12, 11]
12	**Software Effectiveness Evaluation**	Effectiveness is a degree to which something is successful in producing a desired result; success.	[12, 20]
13	**Flexibility**	The capability of secure software to respond to potential internal or external changes affecting its value within timely and cost-effective manner.	[12, 26]
14	**Operational Controls**	The most difficult task of management concerns monitoring the behavior of individuals, comparing security performance to some standard, and providing rewards as specified.	[12, 19]

From the foregoing discussion, researcher classified the security durability attributes into three main levels, the first level, second level and third level attributes which depend on the security durability, directly or indirectly. These attributes helps researcher to assess the security durability of the software. There is no framework or mechanism available to estimate the security durability. Further, assessing the security durability is a very crucial task that's why this work is proposing a method to assess the security durability which is based on Fuzzy MCDA technique. i.e. Fuzzy AHP. Step by step whole process of Fuzzy AHP mechanism is discussed in next section.

3.6 Security Durability Assessment Mechanism

Security is one of the most important quality properties of software which is concerned with both end users and developers [29]. Security estimation is playing a key role to improve the quality of software. Durability plays the key role in enhancing the security life span [12-13]. To improve the security life span of software, security durability assessment is essential that may be helpful to security policy, goals etc. and user's satisfaction. Security cannot be durable until security durability is not measured. Here, the measurement of security durability comprises in two steps including mechanism selection and description & implementation.

3.6.1 Mechanism Selection and Description

It is found that integrating security durability within design may enhance the potential of CIA [24]. Security of a software product is durable, if security works efficiently for user's satisfaction up to expected duration. Further, establishing a relation between durability and security is very crucial. Identification and classification of security durability attributes help to assess and improve security during software development. In order to develop durable as well as secure software, the relationship between security and durability characteristics (at different levels) has been determined in previous sections of this chapter. Expected life span or durability of security is affected by many direct and indirect factors. Direct factors include dependability, trustworthiness, and human trust whereas indirect factors include reliability, consumer integrity, etc. Paying explicit attention on these attributes during early stage of software development may enhance durability of secure software. In addition, assessment of security durability is necessary for security assurance. The assessment process involves not only quantifying attributes of security that contribute towards durability, but also identifying the most crucial attributes among them [12-13, 49]. In a nutshell, identifying, prioritizing, and evaluating the factors are very critical process.

After evaluating the problem of security durability assessment, it is found that this is a decision-making problem which is having multiple criteria in the form of security durability attributes. Thus, in technical terms, security durability assessment relates multiple criteria decision-making problem. There are multiple methods and techniques to solve the problem of decision making [32]. After literature review of previous work, researcher have found that there are so many techniques to solve this type of problems. Further, Multi Criteria Decision Analysis (MCDA) approach is a discipline which aims to support experts when they are faced with various conflicting items for evaluation [97-98]. The MCDA approach is very suitable to take two or more conflicting problems side by side. Various MCDA methods are available including Analytic Hierarchy Process (AHP) and Fuzzy Analytic Hierarchy Process (Fuzzy AHP) etc. [75]. The approaches defined above are differ by their decision are, objective or subjective. This work is using the Fuzzy AHP for assessing the security durability. Further, the results help to formulate development strategies to achieve the desired security durability of software. This may help software developers to come up with durable as well as secure software.

3.6.2 Implementation

In order to address the fundamental difficulty for security durability assessment, researcher have proposed a hybrid method. i.e. Fuzzy AHP methodology [81]. Although, AHP is considered good while analyzing a decision in a group, but various researchers have found that hybrid AHP is better to provide crisp decisions with their weights too [76-77]. Hence, in order to deal with the uncertainty and ambiguity of researchers and academicians, the authors have used a hybrid version of AHP (also known as fuzzy AHP) which incorporates fuzzy set theory with AHP methodology [81], to evaluate security durability of software. The adopted methodology is given in the figure 3.6.2 (a) that is in the form of a flow chart. The flow chart describes the process of security durability assessment. It has been divided into five phases/steps including planning; fuzzification; fuzzy operations; defuzzification; and analysis, confirmation & estimation. Planning phase deals with the problem recognition, selecting the alternatives for the problem and define scope & boundaries of the analytic hierarchy process. Fuzzification phase deals with the preliminary process of methodology including defining the membership function with a scale. Fuzzy operations phase deals with performance of pair wise comparison matrixes through triangular fuzzy numbers with the help of the expert's opinions. Defuzzification phase deals with transformation of fuzzified weights into defuzzified linguistic values while last phase deals with weights, ratings and assessment. Further, last phase also deals with improvement (performance), sensitivity analysis and validation of the results through statistical analysis. The phase wise description of the methodology is given in sub sections as:

a) Planning Phase

The problem of security durability is recognized, addressed in previous chapters and related attributes of security durability are identified, categorized in previous sections of this chapter. AHP is used as a decision-making tool for estimating the priority numbers for different alternatives with hierarchical structure of multiple criteria [99]. According to this research, AHP is best suited for choosing the best alternatives among the number of options while fuzzy is best in dealing with linguistic variables. That's why Fuzzy AHP is used in this work for better results.

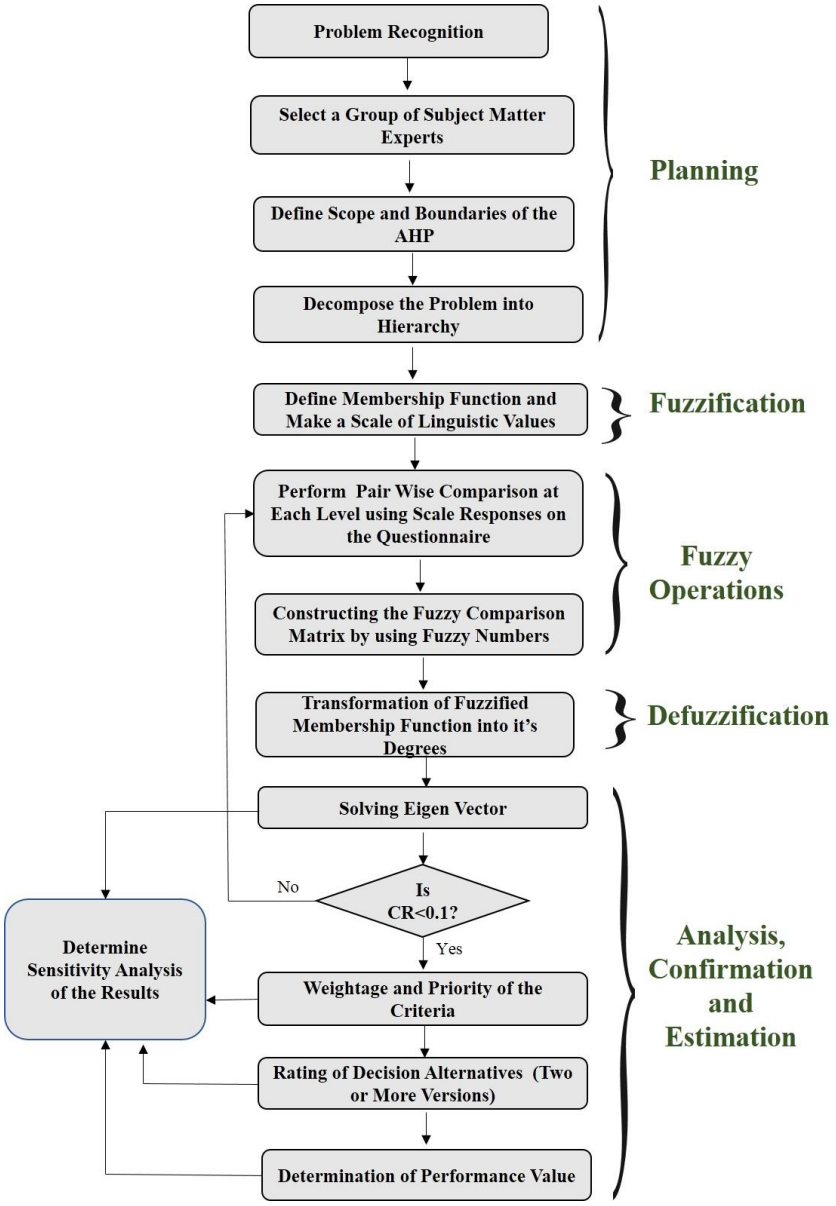

Figure 3.6.2 (a): Flow Chart of the Implementation through Fuzzy AHP Method

b) Fuzzification Phase

To understand the Fuzzy Analytical Hierarchy Process (Fuzzy AHP) methodology, researcher has included a short introduction of both methods and hybridization of them. Saaty defines Analytic Hierarchy Process (AHP) as a decision method which decomposes a complex multi criteria decision problem into a hierarchy [100]. Major benefit of AHP is its relative simplicity with which it handles multiple criteria. AHP allows decision makers to mould a complex problem in a hierarchical structure that consists of the goal, aims, sub-objectives, and alternatives. Traditional methods of AHP cannot be used when there is uncertainty in data. To address such uncertainties, the fuzzy set theory was merged in to the AHP. In 1965, Zadeh

introduced the fuzzy set theory to deal with the uncertainty due to imprecision and vagueness [101]. A fuzzy set is a class of objects with a graded continuum of membership. Such a set is characterized by a membership function, which assigns to each object a membership grade between zero and one. In order to simplify the fuzzy AHP method for this research from the feasible viewpoints, the Fuzzy AHP based on the fuzzy interval arithmetic with triangular fuzzy numbers have been proposed.

In context of the problem addressed in the present work Fuzzy AHP has been used for prioritizing security durability attributes. Triangular fuzzy number help the decision maker to make easier decisions [102]. Hence in this paper triangular fuzzy numbers is used as a membership function. Figure 3.6.2 (b) depicts a triangular fuzzy number.

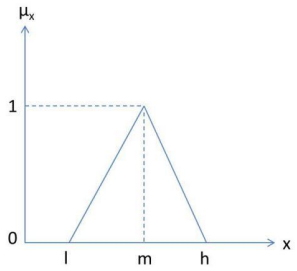

Figure 3.6.2 (b): Triangular Fuzzy Number

In this figure, μ_x is denoted as a membership function where μ denotes membership value of corresponding x. The parameters, l, m and h denote the smallest possible value, the most promising value and the largest possible value, respectively that describes a fuzzy event. Further, a Triangular Fuzzy Number (TFN) (μ_{ij}) is simply denoted as (l, m, h). The triangular fuzzy number μ_{ij} is represented in equation (1):

$$\mu_{ij} = (l_{ij}, m_{ij}, h_{ij}) \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

$$where\ l_{ij} \leq m_{ij} \leq h_{ij} \text{ and } l_{ij}, m_{ij}, h_{ij} \in [1/9, 9]$$

$$l_{ij} = min(B_{ijk}),$$

$$m_{ij} = (B_{ij1}.B_{ij2} \ldots\ldots\ldots\ldots B_{ijk})1/k$$

$$and\ h_{ij} = max(B_{ijk})$$

Where B_{ijk} represents the judgment of experts k for the importance of two criteria $i.e. C_i and\ C_j$

Since each number in the pairwise comparison matrix represents the subjective opinion of decision makers and is an ambiguous concept, fuzzy numbers work best to consolidate fragmented expert opinions [57, 76]. Saaty proposed pair-wise comparisons to create the fuzzy judgment matrix that is used in AHP technique [100] and shown in equation 2.

$$A = [a_{ij}] = \begin{array}{c} \\ C_1 \\ C_2 \\ \vdots \\ C_n \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \dots\ \dots & C_n \end{array} \\ \begin{bmatrix} 1 & a_{11} \dots\dots & a_{1n} \\ 1/a_{21} & 1 \dots\dots & a_{2n} \\ . & . & . \\ . & . & . \\ 1/a_{n1} & 1/a_{n2} \dots\dots & 1 \end{bmatrix} \end{array} \quad \dots\dots\dots\dots(2)$$

Where i = 1,2,3……n and j = 1,2,3……n and a_{ij}= 1:when i=j; and $a_{ij}=1/a_{ij}$; when i≠j

where $[a_{ij}]$ denotes a triangular fuzzy number for the relative importance of two criteria C_i and C_j. Corresponding linguistic scale for membership functions (1 to 9) is given in the table 3.6.2 (a).

Table 3.6.2 (a): Corresponding Linguistic Scale for Membership Functions

S. No.	Linguistic Values	Numeric Values	Fuzzified Numbers (TFNs) $[a_{ij}]$	1/ $[a_{ij}]$
1	Equal Important (Eq)	1	(1,1,1)	(1,1,1)
2	Intermediate Value between Equal and Weakly (E & W)	2	(1,2,3)	(1/3,1/2,1)
3	Weakly Important (WI)	3	(2,3,4)	(1/4,1/3,1/2)
4	Intermediate Value between Weakly and Essential (W & E)	4	(3,4,5)	(1/5,1/4,1/3)
5	Essential Important (EI)	5	(4,5,6)	(1/6,1/5,1/4)
6	Intermediate Value between Essential and Very Strongly (E & VS)	6	(5,6,7)	(1/7,1/6,1/5)
7	Very Strongly Important (VS)	7	(6,7,8)	(1/8,1/7,1/6)
8	Intermediate Value between Very Strongly and Extremely (VS & ES)	8	(7,8,9)	(1/9,1/8,1/7)
9	Extremely Important (ES)	9	(7,9,9)	(1/9,1/9,1/7)

Table 3.6.2 (a) shows the linguistic values into numeric values and numeric values into TFN values. TFN values may be used for creating the pair wise comparison matrix of relative criteria, where a_{ij} denotes the relative importance of criteria *i* comparison with criteria *j* in the scale. To determine the weights of each set of attributes, this scale is used in assessment. Further, the decision made by many experts for security durability is summarized as fuzzy pair wise comparison matrixes. It is also used for characterizing the pair-wise fuzzy judgment matrix which is used in AHP technique. For determining the alternatives importance, linguistic rating scale has been shown in table 3.6.2 (b).

Table 3.6.2 (b): Linguistic Rating Scale

S. No.	Linguistic Value	Numeric Value of Ratings	Fuzzified Ratings (TFNs)
1	Very Low (VL)	0.1	(0.0, 0.1, 0.3)
2	Low (L)	0.3	(0.1, 0.3, 0.5)
3	Medium (M)	0.7	(0.5, 0.7, 0.9)
4	High (H)	0.9	(0.7, 0.9, 1.0)
5	Very High (VH)	1.0	(0.9, 1.0, 1.0)

Table 3.6.2 (b) shows the rating scale of 0 to 1 in scale as 0.1 describes Very Low (VL), 0.3 describes Low (L) and so on. The associated fuzzy values are assigned to every data got from expert for a particular alternative. The process of assessment starts with collecting data by the different number of experts. Data can be collected in forms of questionnaires, checklist, etc. The data acquired from the decision makers are compared pair wise to evaluate the relative importance of each criteria, or the degree of preference of one factor to another with respect to each criteria. However, the perception and judgments of human are represented by linguistic and vague for a complex problem [103-104].

c) Fuzzy Operations

After, various linguistic data is converted into quantitative data into TFN values. To confine the vagueness of the parameters which are related, alternatives such as triangular fuzzy numbers are used [105]. To aggregate the all data into a single form, fuzzy operations are required. If, two TFNs $M_1 = (l1, m1, h1)$ and $M_2 = (l2, m2, h2)$ are given

Then, the rules of operations on them are given below in equation 3, 4 and 5.

$$(l_1, m_1, h_1) + (l_2, m_2, h_2) = (l_1 + l_2, m_1 + m_2, h_1 + h_2) \ldots\ldots\ldots\ldots\ldots(3)$$

$$(l_1, m_1, h_1) \times (l_2, m_2, h_2) = (l_1 \times l_2, m_1 \times m_2, h_1 \times h_2) \ldots\ldots\ldots(4)$$

$$(l_1, m_1, h_1)^{-1} = \left(\frac{1}{h_1}, \frac{1}{m_1}, \frac{1}{l_1}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(5)$$

These fuzzy operations are used in various research areas for decision making in different fields such as decision making, rating and so on [97]. Further, it is based on the rationality of uncertainty due to imprecision. A major contribution of fuzzy set theory is its capability of dealing with uncertainty.

d) Defuzzification

After the construction of comparison matrix, defuzzification is performed to produce a quantifiable value based on the calculated TFN values. The defuzzification method adopted in this work has been derived from [102] as formulated in equations (6-9) which is commonly referred as the alpha cut method.

$$\tilde{A} = [\tilde{a}_{ij}] = \begin{array}{c} \\ C_1 \\ C_2 \\ \vdots \\ C_n \end{array} \begin{array}{c} \begin{array}{ccc} C_1 & C_2 \ldots\ldots & C_n \end{array} \\ \begin{bmatrix} 1 & \tilde{a}_{11} \ldots\ldots & \tilde{a}_{1i} \\ 1/\tilde{a}_{21} & 1 \ldots\ldots & \tilde{a}_{2i} \\ . & . & . \\ . & . & . \\ 1/\tilde{a}_{j1} & 1/\tilde{a}_{j2} \ldots\ldots & 1 \end{bmatrix} \end{array} \ldots\ldots\ldots\ldots\textbf{(6)}$$

Matrix $\tilde{\mathbf{A}}$ is defined as the defuzzified AHP. Where $[\tilde{a}_{ij}]$ denotes a triangular fuzzy number and shows the relative importance between two criteria Ci and Cj. There are different defuzzification methods are available in the literature such as centroid, center of sums, alpha cut etc. [100-105]. In this work, researcher used the alpha cut method for defuzzification. Alpha cut enables one to describe a fuzzy set as a composition of crisp sets. Crisp sets simply describe whether an element is either a member of the set or not. To defuzzify fuzzy matrix ($\tilde{A}$) into crisp matrix ($\rho_{\alpha,\beta}$) is shown in (7-9) (alpha cut method).

$$\rho_{\alpha,\beta}(\tilde{a}_{ij}) = [\beta.\eta_{\alpha}(l_{ij}) + (1-\beta).\eta_{\alpha}(h_{ij})] \quad \text{.....(7)}$$

where $0 \leq \alpha \leq 1$ *and* $0 \leq \beta \leq 1$

such that,

$$\eta_{\alpha}(l_{ij}) = (m_{ij} - l_{ij}).\alpha + l_{ij} \quad \textbf{(8)}$$

$$\eta_{\alpha}(h_{ij}) = h_{ij} - (h_{ij} - m_{ij}).\alpha \quad \textbf{(9)}$$

In equations (7-9), $\eta_{\alpha}(l_{ij})$ denotes the left-end boundary value of alpha cut for $\tilde{a}_{ij}$ and $\eta_{\alpha}(l_{ij})$ denotes the right-end boundary value of alpha cut for $\tilde{a}_{ij}$ Further, α and β carry the meaning of preferences and risk tolerance of participants. Particularly, α and β can be stable or in a fluctuating condition. These two values range between 0 and 1, in such a way that a lesser value indicates greater uncertainty in decision making. Meanwhile the value of α comes to a stable state when it is increasing particularly. Additionally, α and β can be any number between 0 and 1, and analysis is normally set as the following 10 numbers, 0.1, 0.2, up to 0.9 for uncertainty emulation. Since preferences and risk tolerance are not the focus of this contribution, value of 0.5 for α and β is used to represent a balanced value. This indicates that attributes are neither extremely optimistic nor pessimistic about their comparison. Variation due to value of α and β is discussed in sensitivity analysis section. Although, the single pair wise comparison matrix is shown in equation 10.

$$\rho_{\alpha,\beta}(\tilde{A}) = \rho_{\alpha,\beta}[\tilde{a}_{ij}] = \begin{array}{c} \\ C_1 \\ C_2 \\ \vdots \\ C_n \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \ldots\ldots\ldots\ldots & C_n \end{array} \\ \begin{bmatrix} 1 & \rho_{\alpha,\beta}(\tilde{a}_{11}) \ldots\ldots & \rho_{\alpha,\beta}(\tilde{a}_{1i}) \\ 1/\rho_{\alpha,\beta}(\tilde{a}_{21}) & 1 \ldots\ldots & \rho_{\alpha,\beta}(\tilde{a}_{2i}) \\ . & . & . \\ . & . & . \\ 1/\rho_{\alpha,\beta}(\tilde{a}_{j1}) & 1/\rho_{\alpha,\beta}(\tilde{a}_{j2}) \ldots\ldots & 1 \end{bmatrix} \end{array} \quad \text{.....(10)}$$

After defuzzification, to validate the consistency of the matrix, next portion of the section is discussed.

e) Analysis, Confirmation and Estimation:

The next step is to determine the eigenvalue and eigenvector of the fuzzy pair wise comparison matrix. The purpose of calculating the eigenvector is to determine the aggregated weightage of particular criteria. Assume that W denotes the eigenvector, I denotes unitary matrix while λ denotes the eigenvalue of fuzzy pair-wise comparison matrix $\tilde{A}$ or $[\tilde{a}_{ij}]$.

$$[(\rho_{\alpha,\beta} \times \tilde{A}) - \lambda \times I].W = 0 \text{.....(11)}$$

Where $\tilde{A}$ is a fuzzy matrix containing fuzzy numbers of the $\rho_{\alpha,\beta}(\tilde{A})$. Formula (11) is based on the linear transformation of vectors. By applying equations (1-11), the weightage of particular criteria with respect to all other possible criteria can be acquired. The eigenvectors of associated attributes of security durability were then calculated using formula (11) as shown in equation 12.

$$[(\rho_{\alpha,\beta} \times \tilde{A}) - \lambda \times I].W = \begin{bmatrix} 1 & \rho_{\alpha,\beta}(\tilde{a}_{11}) \dots\dots & \rho_{\alpha,\beta}(\tilde{a}_{1i}) \\ 1/\rho_{\alpha,\beta}(\tilde{a}_{21}) & 1 \dots\dots & \rho_{\alpha,\beta}(\tilde{a}_{2i}) \\ . & . & . \\ . & . & . \\ 1/\rho_{\alpha,\beta}(\tilde{a}_{j1}) & 1/\rho_{\alpha,\beta}(\tilde{a}_{j2}) \dots\dots & 1 \end{bmatrix} \text{.....(12)}$$

Multiplying eigenvalue λ with unitary matrix I produce an identity matrix that cancels out each other. Thus, the notation λI is discarded in this case. Applying formulas (11-12) results are shown in equation 13.

$$\begin{bmatrix} 1 & \rho_{\alpha,\beta}(\tilde{a}_{11}) \dots\dots & \rho_{\alpha,\beta}(\tilde{a}_{1i}) \\ 1/\rho_{\alpha,\beta}(\tilde{a}_{21}) & 1 \dots\dots & \rho_{\alpha,\beta}(\tilde{a}_{2i}) \\ . & . & . \\ . & . & . \\ 1/\rho_{\alpha,\beta}(\tilde{a}_{j1}) & 1/\rho_{\alpha,\beta}(\tilde{a}_{j2}) \dots\dots & 1 \end{bmatrix} X \begin{bmatrix} W1 \\ W2 \\ . \\ . \\ . \\ Wn \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ . \\ 0 \end{bmatrix} \text{.....(13)}$$

The aggregated results in terms of weights are shown in equation 13.

In order to control the results of the method, the Consistency Ratio (CR) for each of the matrixes for the hierarchal structure is calculated with the help of equation 14.

$$CR = \frac{CI}{RI} \text{.....(14)}$$

Where, Consistency Index denotes as CI and Random Index denotes as RI [100]. Further, CI is calculated from the equation 15.

$$CI = \frac{\lambda}{(n-1)} \text{.....(15)}$$

Where, n denotes the number **of total responses** and RI is given by Saaty [100] and shown the rank of matrix in table 3.6.2 (c).

Table 3.6.2(c): Random Index

N	1	2	3	4	5	6	7	8	9
Random Index (RI)	0.00	0.00	0.58	0.90	1.12	1.24	1.35	1.41	1.49

With the help of equation 14, 15 and table 3.6.2 (c), CR is calculated. If, CR< 0.1, the approximation is accepted and results are evaluated after this with the help of equation 13; otherwise, a new comparison matrix is solicited.

After calculating the independent weights, this work evaluates the dependent weights and ranks through the hierarchy and results of the obtainable weights gives some suggestion for developers to improve the security durability life span of software services. To assess the effectiveness of results, this work takes two alternatives (version 1 and version 2). Design of version 1 is original from the organization and design of version 2 is changed according to the researcher's suggestions. Through the hierarchy, researcher estimates the independent (that are given in linguistic forms by the designers) and dependent ratings [106-108] of security durability attributes (for version 1 and version 2 respectively) with the help of equations 1, 3, 4, 5 and 7, 8, 9. Then, author assessed the security durability of both alternatives. Overall security durability is assessed by the equation 16 [108-109].

Security Durability = $R_1 \times W_1 + R_2 \times W_2 + \ldots\ldots R_n \times W_n = \Sigma R_i \times W_i$ …… (16)

Or

$$[R_1 \quad R_2 \quad R_3 \ldots . R_i] * \begin{bmatrix} W_1 \\ W_2 \\ W_i \end{bmatrix} = = \Sigma \, R_i \times W_i$$

Where R denotes the rating values, W denotes the weight of associated attribute and i denotes the number of attributes that affect the security durability. The results is cleared the impact of researcher's suggestions and this research work. Further, sensitivity analysis is performed to check the variations on results due to value of α and β.

3.7 Relevant Findings

Relationship between security and durability are crucial, but important process for increasing the lifespan of security for software services. Following are some findings obtained during this chapter:

- Defined security durability in software perspective
- Identified and investigated potential sources to enhance durability of security during use of software services
- Identified, classified, and highlighted the security as well as durability attributes that may harm lifespan of secure software
- Identified and classified the relationship between security and durability
- Identified the co-factors of security durability in different levels
- Arranged them based on the degree of the negative or positive impacts
- Stabilized the relation between security and durability properties
- Detected the best processes of the security durability estimation
- Selected the fuzzy MCDM approach to security durability estimation
- Defined the whole process step by step
- Pinpointed the necessity of integrating the security and durability within the software development life cycle

- Focused on the impact of different security risks that have a great impact on security design, directly or indirectly.

3.8 Conclusion

Prioritization of security durability attributes play an important role to help software developers to focus on fulfilling higher-priority attributes to reduce maintenance cost and time. Because the development of security durability is still in its infancy, there are very limited references and established security durability estimation methodologies that can be adopted by software developers [190]. Software developers need to focus on capturing and prioritizing essential durability attributes so that the time and cost incurred on maintenance can be reduced. The projected work first prepares a strong theoretical foundation for security durability quantification by Fuzzy AHP in this area. It may help researchers to find out the factors related to security durability [235]. A strong correlation has been established between security attributes and durability attributes.

CHAPTER – IV
DEVELOPMENT OF SECURITY DURABILITY ASSESSMENT FRAMEWORK

4.1 Introduction

Security and durability must be combined and assessed from the very beginning of the software development to improve the life span of software services with reduced maintenance cost and time. Features of security and durability must be tested and verified before the application is delivered to user's end. By developing the software with the help of appropriate approach that incorporates security durability, development cost can be more accurately defined and controlled [4, 25]. It also reduces maintenance cost and time that can enhance the security services. Under the aegis of this research work, it is aimed to explore the possibilities for developing a measure to estimate the security durability in the early stage of the software development life cycle in order to maximize the life span of security at an overall level. Basic idea is to assess security durability of software at design phase and optimize the maintenance time and cost at the earliest without any delay. Therefore, a mechanism for ensuring security durability is to be developed in early stage of development process, which may facilitate with knowledge of security durability at design phase [20, 25]. As the design phase produces the structure of the software, making changes and corrections in this phase are much easier than to make them in the subsequent phases. So, there is need to develop an appropriate framework for security durability assessment at this phase. The framework may assist in developing and validating security durability.

Every field such as financial, academics, communication is dependent on different kind of software systems. Increased use of services of software also increases the process of maintenance invariably. This further gives pressure to developers to lessen the cost and time incurred on maintenance. So that user may not face any problem in having continuous services of software [98]. As a solution to this problem developers are trying to develop secure as well as durable software. To facilitate the developers, authors are proposing here a framework for assessment of security durability to improve the working life span of secure software. This framework gives roadmap of assessment of security durability with focus on improving quality

of software. Further, this framework follows some key activities required to simplify and inbuilt security durability into software design. In addition, authors are giving some important recommendations and basic procedure for selecting the best guidelines for fulfilling development program for longer duration of security. In addition, the authors have developed some important recommendations and basic procedure for creating the guidelines to facilitate the development process for having longer and secure software security.

4.2 The Framework

For creating more flexible, usable, durable, and secure software, organizations are always focusing on new ideas to gain trust of the users. Organizations wish to design more secure software which provides longer services to increase user satisfaction. Unfortunately, faster development pace and lack of documentation in software inhibit them to achieve the target of durable security. During the literature survey, the authors found that the existing methodologies to develop secure software are either theoretical or just naive practices. Most of the organizations which are targeting to achieve security of the software somehow ignore longevity of the same. However, focusing on security and durability of the software simultaneously will satisfy the user's needs and investment in secure software [12-13]. Hence, finding ways to develop secure and durable software is still a challenging task.

This is fact that integrating security durability at development phases will reduce cost and rework [54]. Most of the experts have focused on deployment phase of the software development for improving security durability and minimizing maintenance time and cost. But still security durability is not properly achieved. However, researchers and practitioners have advocated integrating security during development phases but nobody has provided the step by step procedure to integrate it with development phases for improving security of software. In addition, no research is seen talking about integrating durability during development phases of software. Hence, there is a need to develop on approach which provides step by step guidelines on how to integrate security durability at the development. Keeping the need in mind, the authors have developed a Security-Durability Framework (SD^f) which provides complete guidance for integrating security durability during development.

4.2.1 Premises

A framework is a schematic representation of a complex process. It provides a step to step guide to perform a task for research. This framework is a living document and can be updated and modified time to time as per the user's security needs. This framework is a common approach to assess security durability of software [110]. To ensure security durability, the framework has the following assumptions:

- The framework improves the life span of security during service life of software with focus on reducing maintenance efforts (time and cost).
- The list of security durability attributes are modifiable during the process of framework implementation. One can choose a subset of the given set of security durability attributes or can also add more security durability attributes.
- The levels of security durability hierarchy are not final. Due to the changes in the number of attributes it is changeable.

Use of this Security Durability Framework (SD^f) is the next step to improve the security of software for enhancing life span of security by reducing maintenance cost and time.

4.2.2 Generic Guidelines

The proposed framework for security durability assessment comprises of five phases/steps (as shown in figure-4.2.3(a)). These are namely

- Factor Identification
- Classification
- Assessment
- Validation
- Packaging

In the first phase i.e. Factor Identification phase, the relevant security principles, relevant durability factors and levels of attributes are identified. In the next phase, i.e. the classification phase, for each identified durability attributes, it is mapped whether the construct adhere to the

identified security principles in order to improve security durability. In assessment phase, prioritization of security durability attributes is done in order to measure security durability confinement through that construct. Algorithm to compute security durability is devised. The fourth phase involves in the validation of the results or assessment that are developed. Fifth phase i.e; the packaging phase evaluates performances on the basis of validation. After this, guideline for developers during software development life cycle is introduced. Review and revision are common in all phases. In this phase the whole approach is revisited for possible improvement and goes back to its last phase from the current one.

4.2.3 Framework Development

Software development organizations have seen phenomenal growth over the last decade which is continuously growing with rapid developments. The big reason behind this growth is increased usage of software in almost each and every field of humans including defense, business, media, sports, etc. As per user's demand and investment on the software, there is need to develop a durable as well as secure software for longer use. Though there is plenty of research available to integrate security during development but irony is that no research is available to integrate durability and security both throughout software development especially at design phase. The authors have developed a conceptual framework which presents key activities to be performed to increase security durability at design phase as shown in figure 4.2.3(a).

Durability has a significant impact on security of software [11-14] which is shown in previous chapters. The increased duration of security will minimize financial assistance and time needed during software maintenance. The mapping of security and durability factors according to organization's needs and importance provides a significant relationship between security and durability. For assessment of the security durability, some principles have been considered in proposing the framework including some key parameters, i.e. security durability contains a set of factors including Dependability, Trustworthiness, Human-Trust (DTH). The factors have an impact on security durability at the time of security software development. The framework presents a roadmap to identify security and durability factors and mapping of these factors in order to evaluate security durability for enhancing the longevity of security.

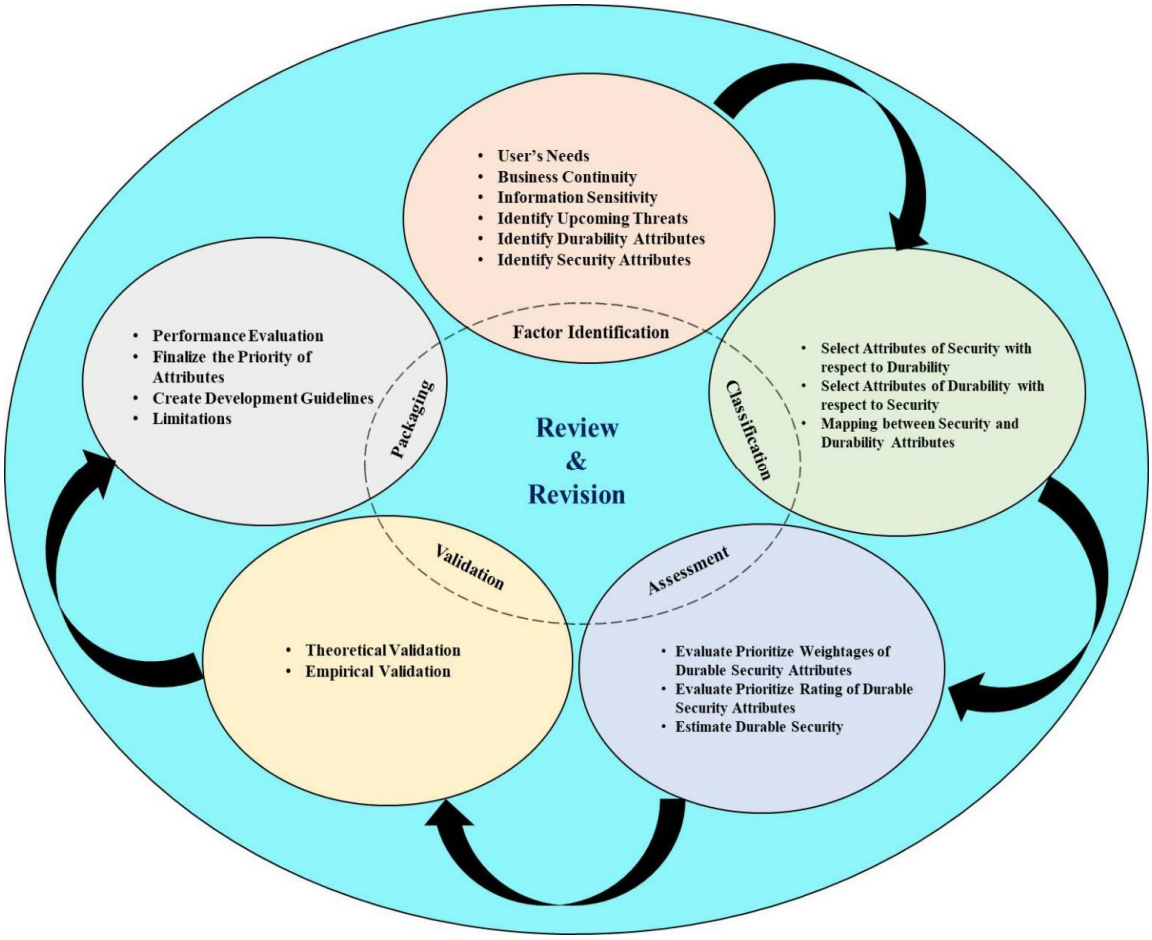

Figure 4.2.3(a): A Framework for Integrating Security Durability Activities at Design Phase

For this, the framework in figure 4.2.3 (a) guides the process of identifying, classifying, evaluating and estimating the security durability. In addition, the framework helps for identifying/selecting the guidelines to integrate all the activities at development level. The security durability framework has five phases including: factor identification; classification; evaluation; validation and finally, result is creation of guidelines with the packaging phase. Review and revision process will be performed as and when required. The upcoming sections discusses about all activities of security durability to be performed in the respective phases.

a) Attribute Identification

Identification of factors is the primary step of most of the problem-solving activities. This stage focuses on the concept of key solutions as well as related facts. The main purpose of this phase is to identify factors of durability and security based on user's needs, information sensitivity, and upcoming threats. User's need refers to the requirement or expectation of user from the security of software. The factors identified here are further treated as key points and help to

make a roadmap for secure and durable design. For factor identification, a pragmatic view should be considered that have a major effect on durability as well as security [210]. Secure and durable design may become complicated, useless and ineffective, if unrelated factors are considered. Hence, only those security and durability factors should be considered and finalized, which also affect the design of software.

b) Mapping between Attributes

In this phase, initially, security attributes with respect to durability is to be classified and vice versa. After that, the mapping or the relationship between security and durability factors should be established. Mapping refers to the embedded relation of one attribute with other and vice versa. This mapping of attributes generates a hierarchy of relation between attributes that will be further helpful for quantification of security durability (durable security). After that, the relationship between security and durability factors is established which is based on best practices and joined set of rules. All the identified factors that can affect the security design must be cross-verified at this part. These durability factors will be addressed in order to control design security. All identified security and durability factors are mapped and established a hierarchy of security durability factors. Here, security durability can be divided into three measurable characteristics including trustworthiness, dependability and human trust.

c) Evaluation

Durability and security are positively related and improving each other simultaneously improves the whole security of software. Enhancing security durability will enhance secure service life of software. Security durability evaluation will help to achieve security goals with lesser cost. The evaluation will also aid to determine factors impact negative or positive on durability as well as security of software. For this, multiple methods may be used, amongst these soft computing-based methods are popular, including AHP (Analytic Hierarchy Process), Fuzzy Analytic Hierarchy Process (Fuzzy AHP), Delphi Analytic Hierarchy Process (Delphi AHP) etc.

Evaluation of security durability supports the improvement of service life of software security. Durability and security seem to have a great even relation. Improving one improvises the other one. It means evaluating durability enhances security and security assessment improves durability as well. Security durability evaluation will also help to crack the goals and minimize the cost spent upon it. A durable security might come up with a more secure and durable system which is also less vulnerable in a life span. In case appropriate results from evaluations are not

achieved, then the fifth step, 'Review & Revision', can be carried out through different techniques and tools, such as the opinions of experts.

d) Validation

The process of validation ensures that the model developed is conforming to the work it is supposed to do. After evaluation of durable security, performance evaluation of security is important. It can be done in two ways, say quantitatively or qualitatively. Though numerical or quantitative assessment is best in this scenario, assessment can be achieved through sensitivity analysis.

The validation process takes care of involved activities to check the building process to produce the right product. The values used in the models are proven as valid measures for the design constructs and it's helpful to examine them in an empirical context. The first step, 'Theoretical validation', is concerned with assuring a theoretical basis through literature study and analysis. The second step, 'Empirical validation', performs a tryout with realistic data to prove that the developed models are valid measures of the desired characteristics. The third step, 'Review & Revision', can be carried out through different means and tools, including opinions of experts and the contextual interpretation and inference of collecting data. The informal reviews and revisions may be carried out at any of the stages in the development process. The last part, 'Finalization', refers to acceptance of valid models and their quantifiable values for security of software.

After evaluation of security durability, performance evaluation of security is important. Performance evaluation is performed to measure the improvement of security after considering durability as a raw factor. Performance evaluation can be done in two ways, either in quantitative or qualitative. Though numerical or quantitative assessment is possible in this scenario, assessment can be achieved through sensitivity analysis. If the results that comes out are not satisfactory then process can go to review and revision phase.

e) Suggestions

Security durability guidelines are the precautionary instructions to be followed to optimize the maintenance time and cost during software is in use. These guidelines will help to calculate the values of security durability to improve security level/ minimize maintenance time and cost of the software. On the basis of assessment, prioritized factors are mapped to select/identify development guidelines. The selection of guidelines is based on the previous proposed

techniques by different practitioners. In this process, software security factors are taken as input and a set of guidelines as output.

Guidelines developed have effects on their associated factors. These effects can either be positive, negative or no effect. To know the affects, mapping of relationships between guidelines and factors is needed. This mapping of relationships helps to identify clashes among the chosen guidelines toward higher-priority factors. To ensure security guidelines should ensure two qualities. Firstly, the guidelines must have positive effects on the high-priority factors. Secondly, the chosen development guidelines should only have homogeneous relationships with each other. The relationships among all chosen development guidelines need to be identified in order to obtain a set of non-overlapping guidelines. On the basis of prioritized factors, development guidelines are created. The algorithm shown in figure 4.2.3(b) is a step by step procedure to produce security durability guidelines.

f) Review & Revision

Review and revision step is common step in all steps, as one can enter into review and revision step in all thorough its development life. This step is informal and as well as important, as review and revision make a framework more manageable. Review and revision, if required, is performed at all the stages to come up with more refined guidelines.

4.3 Framework Significance

The increasing number of incidents on software security breach has imposed the need to look upon a direction to optimize the maintenance time of software development. Also the maintenance of software is more costly to handle as the phases proceed. Hence, the academicians and developers suggest improving the life span of security of the software through design in order to minimize maintenance. Still, there seems a gap between security and durability attributes. Also, no single work has been done to improve software security for duration at the design phase addresses the real problem. No framework for security durability assessment has been identified during the literature survey.

Basic Steps to Create the Set of Guidelines

```
Input       :   Security Durability Factors
Output      :   Set of Guidelines for Improving
                Security Durability

    Let,
        Sf[i] is an array having prioritized factor,
           where i = 0, 1, 2, ……. n
        Sg[j] is an array of guidelines,
           where j = 0, 1, 2, ……. n

 Initially Sg[j]= nil

Step 1: For each prioritized factor ∈ Sf[i]

Create the development guidelines related to factor Sf[i]

Step 1.1:
    if
      guidelines are not conflicting to the set of guidelines in Sg[j]
then
      continue
otherwise
      reject one of them (according to the choice of the security
      designers)

Step 2: For common guidelines of two different factors
Step 2.1:
         if guidelines for Sf[i] match with the guidelines available
            in Sg[j]
        then
       reject the guidelines (chose higher priority factor guidelines)
   else
       store the new guidelines to Sg[j]
```

Repeat these steps for each prioritized factor.

Figure 4.2.3 (b): A Procedure for Creating the Guidelines and Perceptions

That's why the framework developed in this chapter bridges the gap between the software security and software design. It has the twofold advantages: it measures the security durability of software which helps in developing cost effective, durable software. Overall, it enables to answer the questions like 'what is the security durability in the software?', 'which attributes are responsible for less durable software and its security?' At the same time, it also enables to answer the question 'how much security durability has been enhanced? The framework has the following significance:

- It may help to discover and minimize the underlying maintenance in the software at the early stage of software development life cycle leading to a secured end product.

- It may help to determine the effect of the durability over the security during software development process.
- It may assist to develop alternative security versions of durable software under development.
- It may help to assess whether the new versions of two versions of software security is having more security durability than the old one.
- It may help to find out which version is more durable among designs of different software.

4.4 Conclusion

After in-depth literature review and different industry scenarios reveal that maintaining security is tougher than building it. Security durability appears as a milestone in this situation. By the development and application, it can be ensured that security of software might be continuous for expected long life. Although, it is true that ensuring durability of security is complex than developing security itself. All in all, security durability might come up with more security and durability software is less vulnerable software in its life span. In this work, the authors have developed a framework, i.e. security durability assessment framework for integrating durability in security. It provides all the activities required to enhance security durability. With the help of this framework, developers may develop longer secure working life of software to fulfill user's needs. Further, this work also provides the basic ideas for creating development guidelines to facilitate developers to easily maintain CIA for longer duration.

CHAPTER - V
IMPLEMENTATION OF THE FRAMEWORK
-USING FUZZY MULTI CRITERIA DECISION ANALYSIS-

5.1 Introduction

Software security rests upon its attributes including Confidentiality, Integrity, and Availability (CIA) [58]. To meet security needs, developers are trying hard to maintain CIA for longer duration [50-51]. The reason behind this is that there is a huge investment in terms of time, cost, and efforts to develop a secure software [43-45]. Persistence of security for longer duration justifies all the efforts invested in its development. This in turn, reduces maintenance cost as well as time. Hence, a new pillar, in addition to CIA, can be attached to security i.e. durable security (security durability). According to the US Federal report, the software which works with security for longer duration is in demand [111]. Unfortunately, no literature is available to achieve longer security during software development (SDLC). Also, authors found no work (related to improvement through assessment) to enhance security durability at design phase of SDLC. Hence, a well-planned research on security durability assessment during design phase is much needed.

Durability may be expressed as a function of quality in service life of the software [112]. Expected service life or durability of software is affected by many direct and indirect factors. Direct factors include dependability, trustworthiness, usability, sustainability and human trust whereas indirect factors include reliability, consumer integrity etc. Since, dependability, trustworthiness and human trust are three common factors between security and durability [12-13]. Paying explicit attention on these three attributes during early stage of software development may enhance durability of secure software. In addition, assessment of security durability is necessary for security assurance. The assessment process involves not only quantifying attributes of security durability but also identifying the most crucial attributes among them [12]. In a nutshell, identifying, prioritizing, and evaluating the factors is a very critical process [113]. Unfortunately, no attempt has been made for evaluating security durability and balancing their trade-offs in meeting the desired security level. Hence, in this chapter attempts to evaluate and estimate security durability through subjective and objective

assessment with an real time example of security of Entrance Examination Software of Babasaheb Bhimrao Ambedkar University, Lucknow (BBAU Software) [151-152].

A mechanism for security durability assessment is already discussed in chapter 3. According to mechanism, firstly, researcher will evaluate the local weights of security durability attributes through Fuzzy AHP technique (fuzzy method) and put the local weights in the hierarchy and will find the most important attributes in the form of ranks and their final weights. After this, researcher will give suggestions/guidelines for the developers to improve the security life span of software services. To evaluate the security durability of software and impact of the suggestions, researchers will take two versions of BBAU software say version 1 and version 2 where, design of version 1 is based on the organizations (called old version) and design of version 2 is modified, according to the given suggestions (called modified version). To assess the best alternative, the ratings of version 1 and version 2 will be evaluated through fuzzy average method [134]. With the help of weightages (also called subjective weights) and ratings (also called objective weights) of the attributes, overall security durability of version 1 and version 2 is estimated. The step by step process of assessment is shown in next portion of the chapter.

5.2 Evaluating the Weights of Attributes through Fuzzy Method

Through the previous discussion and literature studies it is found that integrating durability within design may enhance the potential of CIA [12]. Hence, firstly establishing a relation between durability and security is important. Security of a software product is durable, if it works efficiently for user's satisfaction up to expected duration. Identification and classification of security durability attributes help to improve security during software development. In order to develop durable as well as secure software, the relationship between security and durability characteristics (at different levels) have been determined in chapter 3, shown in figure 3.5.4 (a)**.** For using the methodology of Fuzzy AHP, these attributes and sub-attributes are converted into a hierarchy that is shown in figure 5.2 (a).

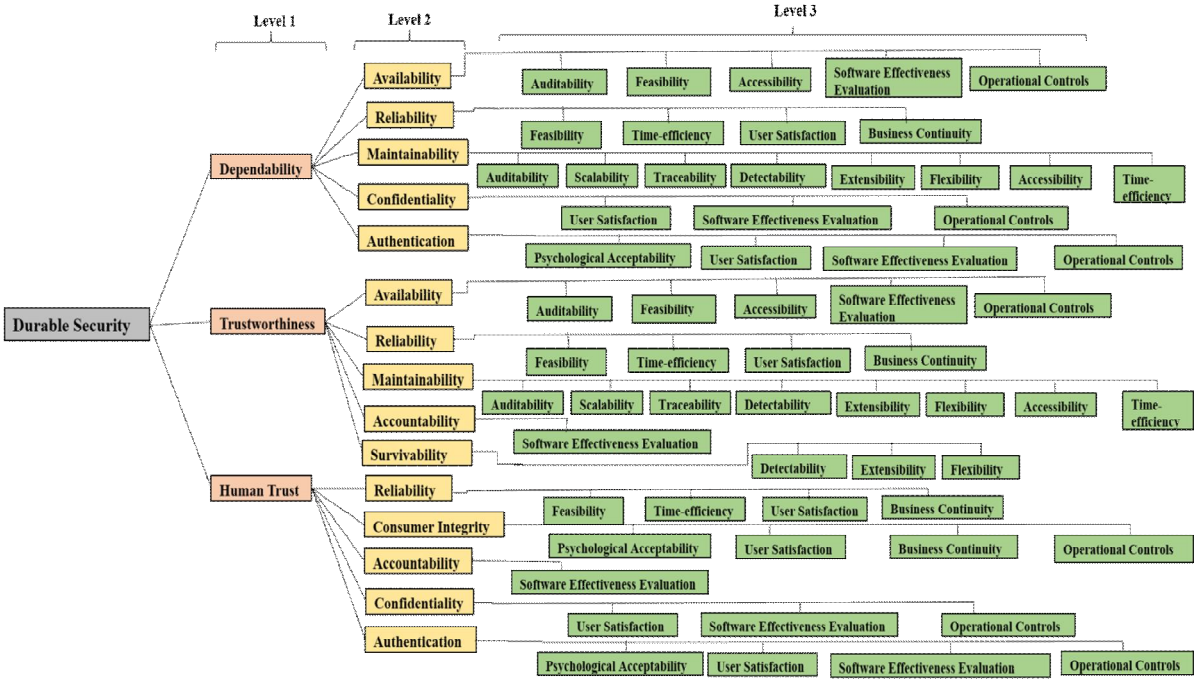

Figure 5.2(a): Hierarchy Modeling of Security Durability Attributes

Figure 5.2 (a) depicts the hierarchical structure of security durability and its attributes which are classified in three levels. At the different level of the hierarchy, the relationship between software quality attributes and software security attributes is shown. Finally, the association of software security attributes with software durability attributes is shown. An attribute at level 1 affects one or more attribute at the higher level but its effect is not same on them, it may vary. For example, reliability has impact on dependability, human trust and trustworthiness as well [114], but its impact values are not same in both levels. Further, the hierarchy of attributes helps to differentiate among the impact of same attribute to the other attribute at the higher level. Among all, the attributes including trustworthiness, human trust and dependability affect the durability directly but many attributes of security affect durability indirectly e.g. availability etc. For the purpose of estimation of security durability, attributes at level 1 are denoted as C1, C2 and C3. Attributes at level 2 are denoted as C11, C12, C13, C14, C15 for C1 and C21, C22……C25 for C2 and C31, C32…..C35 for C3. Attributes at level 3 are denoted as C111…….C115 for C11 and so on which are shown in Figure 5.2 (a).

5.2.1 Construction of Pair Wise Comparison Matrices

Many times, assessment of different attributes usually fails because of the connection of multiple qualitative criteria. Fuzzy AHP is a suitable evaluation technique capable of handling this kind of problem with uncertain inputs. Fuzzy AHP is capable of handling ambiguous

judgmental inputs given by the number of experts and questionnaires collected by judgments of experts. It is also capable of converting qualitative inputs into quantitative results, in form of weightage, ranking, as well as performance. To evaluate the weights of the security durability attributes, pair wise comparison matrixes are constructed in the form of questionnaires for each set of attributes and data has been collected by distributing questionnaires to 50 academicians and industry persons of various affiliations. 20 valid replies were used in this research to measure the importance of security durability attributes.

The data collected through expert's opinions has been arranged in the form of decision matrices. Eigenvector method has been used for taking expert's views. Also, repeated data and redundancy has been removed using a 'data only once' method. Although during calculation, these repetitions have been taken into account as every attribute has different impact on security durability at different levels of hierarchy. To construct the pair wise comparison matrices, table 3.6.2 (a) shown a scale in chapter 3. This scale is a nine-point scale ranging from 1- 9, where a greater value represents higher importance. This scale also helped to convert the numerical values into Triangular Fuzzy Numbers (TFN). TFN's can be obtained for computing the fuzzified values of the linguistic terms from the pair wise judgment matrix. Further, TFN helps to the person in making decision easily. Hence, TFN is used as the membership function in this work.

5.2.2 Aggregation of Pair Wise Comparison Matrices

With the help of table 3.6.2 (a) and equations (1-5) given in mechanism section of chapter 3, researcher converted the numerical values into TFN and aggregated these values. For the all sets of attributes of the hierarchy, aggregated pair wise comparison matrices are shown from table 5.2.2(a) to table 5.2.2(k).

Table 5.2.2(a): Aggregated Fuzzify Pair Wise Comparison Matrix for the First Level

	Dependability (C1)	**Trustworthiness** (C2)	**Human Trust** (C3)
Dependability (C1)	1	1.3479, 1.8180, 2.3859	1.4131, 1.9651, 2.4820
Trustworthiness (C2)	-	1	0.8540, 1.1087, 1.4532
Human Trust (C3)	-	-	1

Table 5.2.2 (a) shows the aggregated fuzzify pair wise comparison matrix of first level attributes including dependability (C1), trustworthiness (C2) and human trust (C3).

Table 5.2.2(b): Aggregated Fuzzify Pair Wise Comparison Matrix for C1 of Second Level

	Availability (C11)	**Reliability** (C12)	**Maintainability** (C13)	**Confidentiality** (C14)	**Authentication** (C15)
Availability (C11)	1	0.3127, 0.4395, 0.6252	0.8733, 0.9012, 0.9465	0.2261, 0.2928, 0.4166	0.2580, 0.3386, 0.5055
Reliability (C12)	-	1	2.0451, 3.1699, 4.2330	0.2665, 0.3657, 0.5911	0.6906, 1.0059, 1.5117
Maintainability (C13)	-	-	1	0.3667, 0.5251, 0.9659	0.3604, 0.5220, 0.8074
Confidentiality (C14)	-	-	-	1	0.8960, 1.1486, 1.3903
Authentication (C15)	-	-	-	-	1

Table 5.2.2 (b) shows the aggregated fuzzify pair wise comparison matrix of second level attributes for dependability including availability (C11), reliability (C12), maintainability (C13), confidentiality (C14) and authentication (C15).

Table 5.2.2(c): Aggregated Fuzzify Pair Wise Comparison Matrix for C2 of Second Level

	Availability (C21)	**Reliability** (C22)	**Maintainability** (C23)	**Accountability** (C24)	**Survivability** (C25)
Availability (C21)	1	0.5598, 0.8994, 1.3705	0.7912, 0.8831, 1.0204	0.4956, 0.7029, 0.9330	0.4067, 0.5497, 0.7876
Reliability (C22)	-	1	0.8001, 1.2376, 1.7812	0.3836, 0.5483, 0.8344	0.4876, 0.6710, 0.8900
Maintainability (C23)	-	-	1	0.5966, 0.7093, 0.9095	0.2770, 0.3854, 0.6340
Accountability (C24)	-	-	-	1	0.5506, 0.5881,

					0.6647
Survivability (C25)	-	-	-	-	1

Table 5.2.2 (c) shows the aggregated fuzzify pair wise comparison matrix of second level attributes for trustworthiness including availability (C21), reliability (C22), maintainability (C23), accountability (C24) and survivability (C25).

Table 5.2.2(d): Aggregated Fuzzify Pair Wise Comparison Matrix for C3 of Second Level

	Reliability (C31)	**Consumer Integrity** (C32)	**Accountability** (C33)	**Confidentiality** (C34)	**Authentication** (C35)
Reliability (C31)	1	0.9710, 1.2475, 1.6094	1.0592, 1.5849, 2.2206	0.7733, 1.0118, 1.2881	0.7612, 0.9120, 1.0965
Consumer Integrity (C32)	-	1	0.6352, 0.9143, 1.3430	0.4273, 0.6335, 0.9660	0.3476, 0.4900, 0.8734
Accountability (C33)	-	-	1	0.5146, 0.6575, 0.7846	0.5213, 0.6597, 0.9191
Confidentiality (C34)	-	-	-	1	0.5562, 0.6448, 0.8122
Authentication (C35)	-	-	-	-	1

Table 5.2.2 (d) shows the aggregated fuzzify pair wise comparison matrix of second level attributes for human trust including reliability (C31), consumer integrity (C32), accountability (C33), confidentiality (C34) and authentication (C35).

Table 5.2.2(e): Aggregated Fuzzify Pair Wise Comparison Matrix for C11 of Third Level

	Auditability (C111)	**Feasibility** (C112)	**Accessibility** (C113)	**Software Effectiveness Evaluation** (C114)	**Operational Controls** (C115)
Auditability (C111)	1	1.8722, 2.5710, 3.2035	1.4640, 1.6842, 1.9743	1.4461, 2.4385, 3.3865	0.4677, 0.5724, 0.7845
Feasibility (C112)	-	1	0.6083, 0.7754, 1.0265	0.7708, 0.9504, 1.2361	0.1630, 0.1953, 0.2497
Accessibility (C113)	-	-	1	0.7694,1. 0502, 1.3553	0.2086, 0.2462, 0.3117
Software Effectiveness Evaluation (C114)	-	-	-	1	0.1956, 0.2283, 0.2903
Operational Controls (C115)	-	-	-	-	1

Table 5.2.2 (e) shows the aggregated fuzzify pair wise comparison matrix of third level attributes for availability (related to dependability) including auditability (C111), feasibility (C112), accessibility (C113), software effectiveness evaluation (C114) and operational controls (C115).

Table 5.2.2(f): Aggregated Fuzzify Pair Wise Comparison Matrix for the C12 of Third Level

	Feasibility (C121)	**Time-efficiency** (C122)	**User Satisfaction** (C123)	**Business Continuity** (C124)
Feasibility (C121)	1	1.7561, 2.3498, 3.0335	1.4830, 1.9575, 2.5293	1.1284, 1.5543, 1.9884
Time-efficiency (C122)	-	1	0.5695, 0.7860, 1.1555	0.5698, 0.7195, 0.9699
User Satisfaction (C123)	-	-	1	0.6270, 0.8123, 1.0718
Business Continuity (C124)	-	-	-	1

Table 5.2.2 (f) shows the aggregated fuzzify pair wise comparison matrix of third level attributes for reliability (related to dependability) including feasibility (C121), time-efficiency (C122), user satisfaction (C123), and business continuity (C124).

Table 5.2.2(g): Aggregated Fuzzify Pair Wise Comparison Matrix for the C13 of Third Level

	Auditability (131)	**Scalability** (132)	**Traceability** (133)	**Detectability** (134)	**Extensibility** (135)	**Flexibility** (136)	**Accessibility** (137)	**Time-efficiency** (138)
Auditability (131)	1	1.0000, 1.5157, 1.9331	0.4896, 0.6372, 1.0000	0.4152, 0.5743, 1.0000	0.2215, 0.2871, 0.4152	0.3146, 0.4610, 0.8705	0.6575, 1.1653, 1.6883	0.2444, 0.3238, 0.4801
Scalability (132)	-	1	0.5743, 0.6657, 0.8022	0.3039, 0.3936, 0.5661	0.2679, 0.3521, 0.5176	0.1663, 0.1969, 0.2531	0.3930, 0.5743, 1.0564	0.1692, 0.2076, 0.2759
Traceability (133)	-	-	1	1.0000, 1.3195, 1.5518	0.3009, 0.4352, 0.8027	0.8027, 0.8705, 1.0000	1.2619, 1.8250, 2.4334	0.1728, 0.2091, 0.2648
Detectability (134)	-	-	-	1	0.5386, 0.9143, 1.5836	0.6083, 1.0592, 1.6829	0.7503, 1.3465, 1.9611	0.6790, 0.7489, 0.8705
Extensibility (135)	-	-	-	-	1	0.4152, 0.6372, 1.1791	0.9465, 1.1095, 1.2457	0.2500, 0.3300, 0.5000
Flexibility (136)	-	-	-	-	-	1	1.8881, 2.5508, 3.1697	0.8027, 1.0352, 1.3160
Accessibility (137)	-	-	-	-	-	-	1	0.2136, 0.2575, 0.3195
Time-efficiency (138)	-	-	-	-	-	-	-	1

Table 5.2.2 (g) shows the aggregated fuzzify pair wise comparison matrix of third level attributes for maintainability (related to dependability) including auditability (C131), scalability (C132), traceability (C133), detectability (C134), extensibility (C135), flexibility (C136), accessibility (C137) and time-efficiency (C138).

Table 5.2.2(h): Aggregated Fuzzify Pair Wise Comparison Matrix for the C14 of Third Level

	User Satisfaction (C141)	**Software Effectiveness Evaluation (**C142)	**Operational Controls** (C143)
User Satisfaction (C141)	1	0.6898, 0.8860, 1.1002	0.2255, 0.2762, 0.3574
Software Effectiveness Evaluation (C142)	-	1	0.3051, 0.3892, 0.5609
Operational Controls (C143)	-	-	1

Table 5.2.2 (h) shows the aggregated fuzzify pair wise comparison matrix of third level attributes for confidentiality (related to dependability) including user satisfaction (C141), software effectiveness evaluation (C142) and operational controls (C143).

Table 5.2.2(i): Aggregated Fuzzify Pair Wise Comparison Matrix for the C15 of Third Level

	Psychological Acceptability C151)	**User Satisfaction** (C152)	**Software Effectiveness Evaluation** (C153)	**Operational Controls** (C154)
Psychological Acceptability (C151)	1	1.0000, 1.3741, 1.7118	0.5610, 0.8360, 1.0781	0.3040, 0.3766, 0.4723
User Satisfaction (C152)	-	1	0.3030, 0.4208, 0.6052	0.1916, 0.2303, 0.3001
Software Effectiveness Evaluation (C153)	-	-	1	0.5138, 0.7959, 1.2032
Operational Controls (C154)	-	-	-	1

Table 5.2.2 (i) shows the aggregated fuzzify pair wise comparison matrix of third level attributes for authentication (related to dependability) including psychological acceptability (C151), user satisfaction (C152), software effectiveness evaluation (C153) and operational controls (C154).

Due to repeated attributes in second level, some set of third level attributes are repeated when set of attributes considered independently. Hence, aggregated fuzzify pair wise comparison matrixes of third level attributes for C21, C22 and C23 (related to trustworthiness) are same as C11, C12 and C13 respectively. According to hierarchy, accountability (C24) depends only on software effectiveness evaluation (C241) with respect to security durability. So, there is no need of fuzzify pair wise comparison matrix. Further, aggregated fuzzify pair wise comparison matrix for the C25 of third level is shown in table 5.2.2 (j).

Table 5.2.2(j): Aggregated Fuzzify Pair Wise Comparison Matrix for the C25 of Third Level

	Detectability (C251)	**Extensibility** (C252)	**Flexibility** (C253)
Detectability (C251)	1	0.6950, 0.9502, 1.3457	1.1486, 1.4385, 1.6962
Extensibility (C252)	-	1	1.1928, 1.5826, 2.1497
Flexibility (C253)	-	-	1

Table 5.2.2 (j) shows the aggregated fuzzify pair wise comparison matrix of third level attributes for survivability (related to trustworthiness) including detectability (C251), extensibility (C252) and flexibility (C253).

Table 5.2.2(k): Aggregated Fuzzify Pair Wise Comparison Matrix for the C32 of Third Level

	Psychological Acceptability (C321)	**User Satisfaction** (C322)	**Business Continuity** (C323)	**Operational Controls** (C324)
Psychological Acceptability (C321)	1	1.07810, 1.5990, 2.1130	0.8206, 1.1118, 1.6150	0.5670, 0.7132, 0.8739
User Satisfaction (C322)	-	1	0.3230, 0.4480, 0.6051	0.2584, 0.3172, 0.4168
Business Continuity (C323)	-	-	1	0.6661, 1.0564, 1.5427
Operational Controls (C324)	-	-	-	1

Table 5.2.2 (k) shows the aggregated fuzzify pair wise comparison matrix of third level attributes for consumer integrity (related to human trust) including psychological acceptability (C321), user satisfaction (C322), business continuity (C323) and operational controls (C324). Again, aggregated fuzzify pair wise comparison matrixes of third level attributes for C31, C34 and C35 (related to human trust) are same as C12, C14 and C15 respectively. Further, accountability (C33) depends only on software effectiveness evaluation (C331) with respect to security durability. So, there is no need for fuzzify pair wise comparison matrix. After the Aggregation of fuzzify pair wise comparison matrixes, defuzzification process is implemented in next portion.

5.2.3 Defuzzification and Local Weights

Now for getting the linguistic values from the aggregated TFN values, the alpha cut method is used for defuzzification process [102]. Alpha Cut method is formulated in equations (6-9) in chapter 3. Example: if TFN of $a_{12} = (l_{12,} m_{12,} h_{12}) = (1.3479, 1.8180, 2.3859)$

then,

$\eta_{0.5}(l_{12}) = (m_{12} - l_{12}).0.5 + l_{12}$

$\eta_{0.5}(l_{12}) = (1.8180 - 1.3479) \times 0.5 + 1.3479 = 1.5830$

$\eta_{0.5}(h_{12}) = h_{12} - (h_{12} - m_{12}). 0.5$

$\eta_{0.5}(h_{12}) = 2.3859 - (2.3859 - 1.8180) \times 0.5 = 2.1019$

$\rho_{0.5, 0.5}(\eta_{12}) = [0.5\ \eta_{0.5}(l_{12}) + (1-0.5)\ \eta_{0.5}(h_{12})]$

$\rho_{0.5, 0.5}(\eta_{12}) = [0.5 \times 1.5830 + (1 - 0.5) \times 2.1019] = 0.7915 + 1.0510 = 1.8425$

$\rho_{0.5, 0.5}(\eta_{21}) = 1/ 1.8425 = 0.5427$

Similarly, all aggregated TFN values defuzzified that are shown from the 5.2.3(a) to 5.2.3(u). In this work, α and β is taken equal to 0.5. Where, α and β carry the meaning of preferences and risk tolerance of participants. The values of α=0.5 and β=0.5 indicated that attributes are neither extremely optimistic nor pessimistic about their comparison. Further, variation in results due to value of α and β is discussed in sensitivity analysis in chapter 7. After defuzzification of pair wise matrix, Consistency Ratio (CR) is calculated with the help of equations (14-15) and table 3.6.2 (c) which are already discussed in chapter 3. To continue the Fuzzy AHP analysis, CR must be acceptable. If CR is less than 0.1 then weights are calculated otherwise refined pair

wise matrixes are prepared and the process is repeated again. After verification of the CR value, by applying equations (12-13), local weightages of security durability attributes are calculated.

For example:

$$[\rho_{\alpha,\beta}(\eta_{ij}) - \lambda I] = \begin{bmatrix} 1 & 1.8425 & 1.9564 \\ 0.5427 & 1 & 1.1312 \\ 0.5111 & 0.8840 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1.8425 & 1.9564 \\ 0.5427 & 1 & 1.1312 \\ 0.5111 & 0.8840 & 1 \end{bmatrix} \begin{bmatrix} \rho \text{ Dependability} \\ \rho \text{ Trustworthiness} \\ \rho \text{ Human Trust} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \rho \text{ Dependability} \\ \rho \text{ Trustworthiness} \\ \rho \text{ Human Trust} \end{bmatrix} = \begin{bmatrix} 0.4867 \\ 0.2698 \\ 0.2435 \end{bmatrix}$$

Similarly, the process is repeated to check the CR and obtain the local weights. From table 5.2.3(a) to 5.2.3(k) and figure 5.2.3(a) to 5.2.3(k) shows the local weights and CR values for each pair wise comparison matrix.

Table 5.2.3(a): Local Weight of Attributes for First Level through Fuzzy Method

	Dependability (C1)	**Trustworthiness** (C2)	**Human Trust** (C3)	**Weights**
Dependability (C1)	1	1.8425	1.9564	**0.4867**
Trustworthiness (C2)	0.5 427	1	1.1312	**0.2698**
Human Trust (C3)	0.5 111	0.8840	1	**0.2435**
				CR= 0.00038

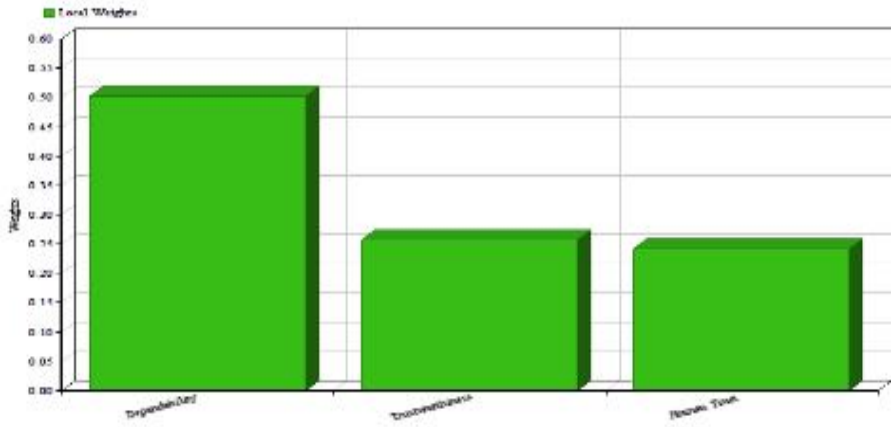

Figure 5.2.3(a): Graphical Representation of Local Weights for First Level through Fuzzy Method

Table 5.2.3(a) and figure 5.2.3(a) shows the local weights of first level attributes of the hierarchy. Consistency Ratio (CR) is 0.00038 and which is less than 0.1. This CR value is acceptable to continue Fuzzy AHP analysis. This set of attributes have three attributes including dependability (0.4867), trustworthiness (0.2698) and human trust (0.2435) and dependability is highest weighted factor among them.

Table 5.2.3(b): Local Weight of Attributes for C1 of Second Level through Fuzzy Method

	Availability (C11)	**Reliability** (C12)	**Maintainability** (C13)	**Confidentiality** (C14)	**Authentication** (C15)	**Weights**
Availability (C11)	1	0.4542	0.9056	0.3071	0.3602	**0.0946**
Reliability (C12)	2.2017	1	3.1545	0.3973	1.0536	**0.2292**
Maintainability (C13)	1.1042	0.31701	1	0.5957	0.5530	**0.1192**
Confidentiality (C14)	3.2563	2.5170	1.6787	1	1.1459	**0.3233**
Authentication (C15)	2.7762	0.9491	1.8083	0.8727	1	**0.2337**
						C.R.=0.0411

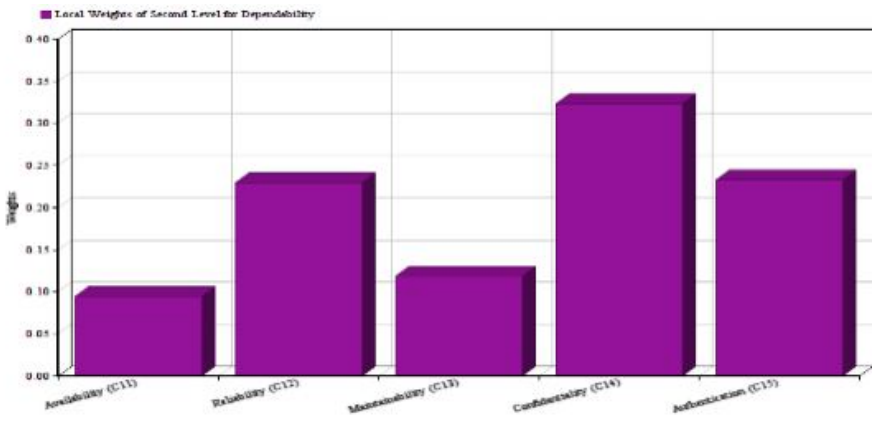

Figure 5.2.3(b): Graphical Representation for C1 of Second Level through Fuzzy Method

Table 5.2.3(b) and figure 5.2.3(b) shows the local weights for C1 of Second Level attributes. Consistency Ratio (CR) is 0.0411 and less than 0.1. This CR value is acceptable to continue Fuzzy AHP analysis. This set of attributes have five attributes including availability (0.0946), reliability (0.2292), maintainability (0.1192), confidentiality (0.3233) and authentication (0.2337) and confidentiality is highest weighted factor among them.

Table 5.2.3(c): Local Weight of Attributes for C2 of Second Level through Fuzzy Method

	Availability (C2	**Reliability** (C2	**Maintainabilit** (C23)	**Accountabilit** (C24)	**Survivability** (C25)	**Weights**
Availability (C21)	1	0.9323	0.8945	0.7086	0.5734	**0.1541**
Reliability (C22)	1.0726	1	1.2642	0.5787	0.6647	**0.1692**
Maintainability (C23)	1.1179	0.7910	1	0.7304	0.4205	**0.1476**
Accountability (C24)	1.4112	1.7280	1.3691	1	0.5979	**0.2214**
Survivability (C25)	1.7440	1.5044	2.3781	1.6725	1	**0.3077**
						C.R.=0.0101

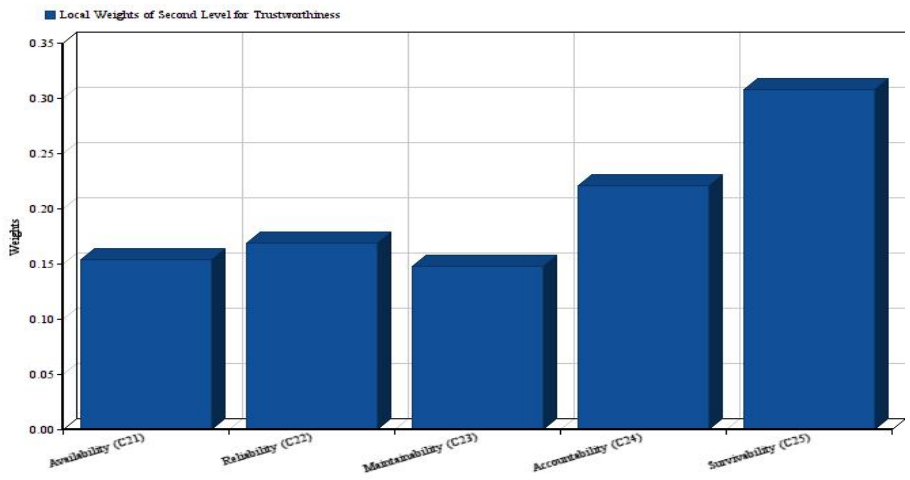

Figure 5.2.3(c): Graphical Representation for C2 of Second Level through Fuzzy Method

Table 5.2.3(c) and figure 5.2.3(c) shows the local weights for C2 of Second Level attributes. Consistency Ratio (CR) is 0.0101 which is less than 0.1. This CR value is acceptable to continue Fuzzy AHP analysis. This set of attributes has five attributes including availability (0.1541), reliability (0.1692), maintainability (0.1476), accountability (0.2214), survivability (0.3077) and survivability is highest weighted factor among them.

Table 5.2.3(d): Local Weight of Attributes for C3 of Second Level through Fuzzy Method

	Reliability (C31)	**Consumer-Integrity** (C32)	**Accountability** (C33)	**Confidentiality** (C34)	**Authentication** (C35)	**Weights**
Reliability (C31)	1	1.2689	1.6124	1.0213	0.9204	**0.2216**
Consumer Integrity (C32)	0.7881	1	1.2693	0.6651	0.5503	**0.1596**
Accountability (C33)	0.6202	0.7878	1	0.6536	0.6900	**0.1446**
Confidentiality (C34)	0.9791	1.5035	1.5300	1	0.6645	**0.2115**
Authentication (C35)	1.0865	1.8172	1.4493	1.5049	1	**0.2627**
						C.R.=0.0069

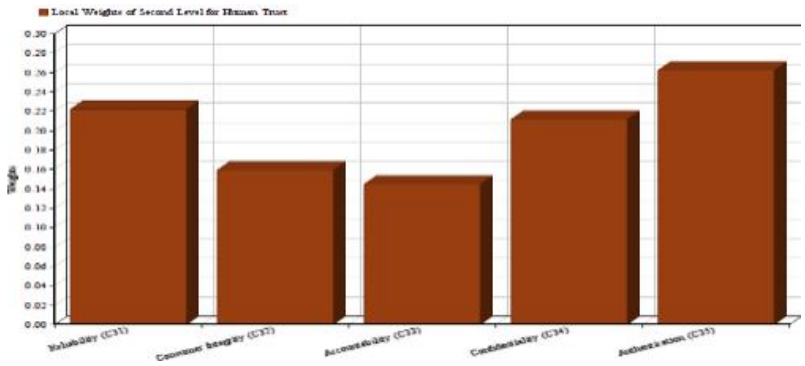

Figure 5.2.3 (d): Graphical Representation for C3 of Second Level through Fuzzy Method

Table 5.2.3 (d) and figure 5.2.3 (d) shows the local weights for C3 of Second Level attributes. Consistency Ratio (CR) is 0.0069 which is less than 0.1. This CR value is acceptable to continue Fuzzy AHP analysis. This set of attributes have five attributes including reliability (0.2216), consumer integrity (0.1596), accountability (0.1446), confidentiality (0.2115), authentication (0.2627) and authentication is highest weighted factor among them.

Table 5.2.3 (e): Local Weight of Attributes for C11 of Third Level through Fuzzy Method

	Auditability (C111)	**Feasibility** (C112)	**Accessibility** (C113)	**Software Effectiveness Evaluation** (C114)	**Operational Controls** (C115)	**Weights**
Auditability (C111)	1	2.5544	1.7017	2.4274	0.5993	**0.2400**
Feasibility (C112)	0.3915	1	0.7964	0.9769	0.2073	**0.0952**
Accessibility (C113)	0.5876	1.2556	1	1.0563	0.2532	**0.1200**
Software Effectiveness Evaluation (C114)	0.4120	1.0236	0.9467	1	0.2357	**0.1032**
Operational Controls (C115)	1.6686	4.8239	3.9495	4.2427	1	**0.4416**
						C.R.=0.0025

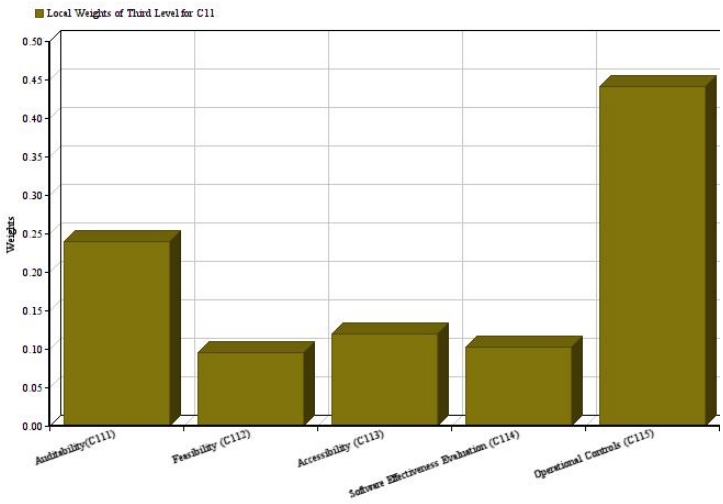

Figure5.2.3 (e): Graphical Representation for C11 of Third Level through Fuzzy Method

Table 5.2.3(e) and figure 5.2.3(e) shows the local weights for C11 of Third Level attributes. Consistency Ratio (CR) is 0.0025 which is less than 0.1. This CR value is acceptable to continue Fuzzy AHP analysis. This set of attributes have five attributes including auditability (0.2400), feasibility (0.0952), accessibility (0.1200), software effectiveness evaluation (0.1032), operational controls and operational controls is highest weighted factor among them.

Table 5.2.3(f): Local Weight of Attributes for C12 of Third Level through Fuzzy Method

	Feasibility (C121)	**Time-efficiency** (C122)	**User Satisfaction** (C123)	**Business Continuity** (C124)	**Weights**
Feasibility (C121)	1	2.3723	1.9819	1.5564	**0.3905**
Time-efficiency (C122)	0.4215	1	0.8243	0.7447	**0.1694**
User Satisfaction (C123)	0.5046	1.2132	1	0.8309	**0.2004**
Business Continuity (C124)	0.6425	1.3428	1.2035	1	**0.2397**
					CR= 0.0006

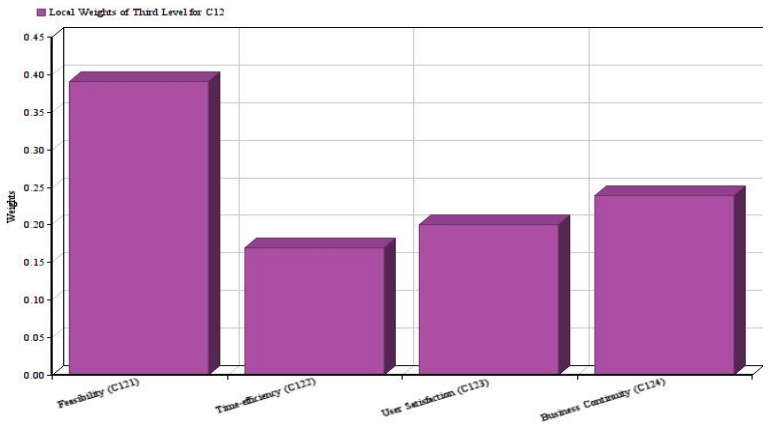

Figure 5.2.3(f): Graphical Representation for C12 of Third Level through Fuzzy Method

Table 5.2.3(f) and figure 5.2.3(f) shows the local weights for C12 of Third Level attributes. Consistency Ratio (CR) is 0.0006 which is less than 0.1. This CR value is acceptable to continue Fuzzy AHP analysis. This set of attributes have four attributes including feasibility (0.3905), time-efficiency (0.1694), user satisfaction (0.2004), business continuity (0.2397) and feasibility is highest weighted factor among them.

Table 5.2.3(g): Local Weight of Attributes for C13 of Third Level through Fuzzy Method

	Auditability (131)	**Scalability (132)**	**Traceability (133)**	**Detectability (134)**	**Extensibility (135)**	**Flexibility (136)**	**Accessibility (137)**	**Time-efficiency (138)**	**Weights**
Auditability (131)	1	1.4912	0.6910	0.6410	0.3027	0.5268	1.1691	0.3430	**0.0733**
Scalability (132)	0.6706	1	0.6770	0.4143	0.3724	0.2033	0.6495	0.2151	**0.0497**
Traceability (133)	1.4470	1.4771	1	1.2977	0.4935	0.8520	1.8364	0.2140	**0.1031**
Detectability (134)	1.5600	2.4137	0.7706	1	0.9636	1.1024	1.3511	0.7319	**0.1271**
Extensibility (135)	3.3036	2.6853	2.0263	1.0378	1	0.7172	1.1028	0.4350	**0.1414**
Flexibility (136)	1.8982	4.9188	1.1737	0.9071	1.3943	1	2.3852	1.0473	**0.1729**
Accessibility (137)	0.8554	1.5397	0.5445	0.7401	0.90679	0.41925	1	0.2621	**0.0760**
Time-efficiency (138)	2.9154	4.6490	4.6729	1.36631	2.2989	0.95484	3.8153	1	**0.2565**
									C.R.=0.0333

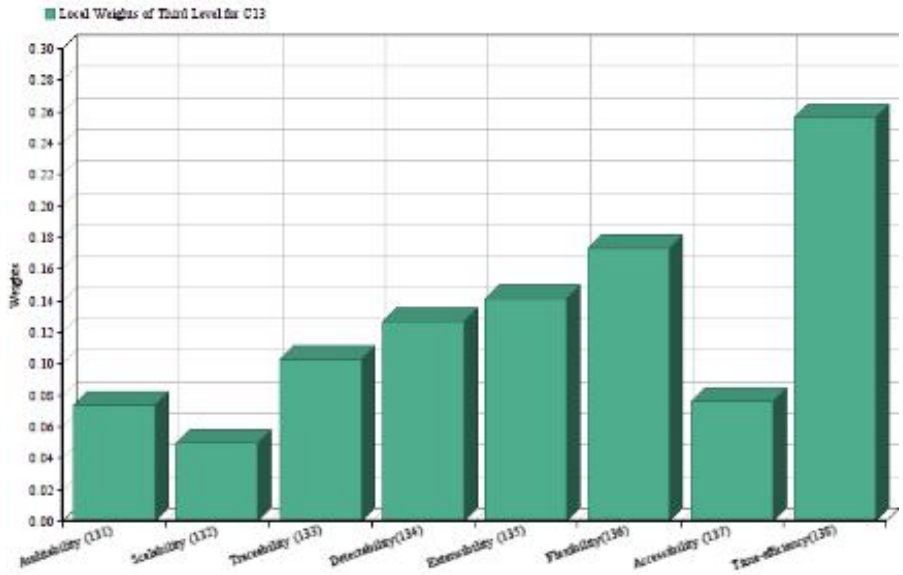

Figure 5.2.3(g): Graphical Representation for C13 of Third Level through Fuzzy Method

Table 5.2.3(g) and figure 5.2.3(g) shows the local weights for C13 of Third Level attributes. Consistency Ratio (CR) is 0.0333 which is less than 0.1. This CR value is acceptable to continue Fuzzy AHP analysis. This set of attributes have eight attributes including auditability (0.0733), scalability (0.0497), traceability (0.1031), detectability (0.1271), extensibility (0.1414), flexibility (0.1729), accessibility (0.0760), time-efficiency (0.2565) and time-efficiency is highest weighted factor among them.

Table 5.2.3(h): Local Weight of Attributes for C14 of Third Level through Fuzzy Method

	User Satisfaction (C141)	**Software Effectiveness Evaluation** (C142)	**Operational Controls** (C143)	**Weights**
User Satisfaction (C141)	1	0.8905	0.2839	**0.1832**
Software Effectiveness Evaluation (C142)	1.1230	1	0.4111	**0.2239**
Operational Controls (C143)	3.5224	2.4325	1	**0.5929**
				CR= 0.0062

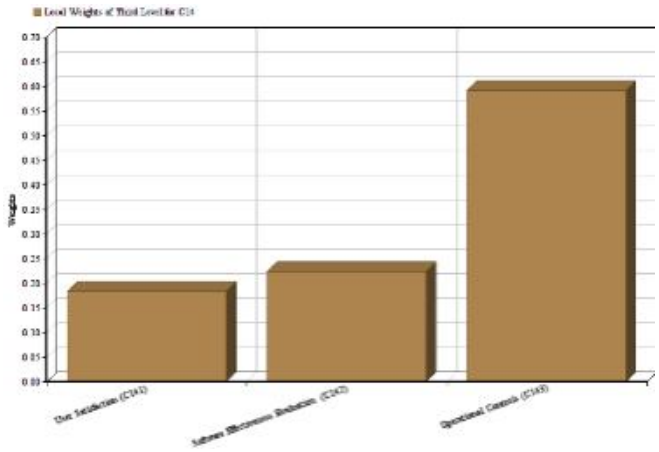

Figure 5.2.3(h): Graphical Representation for C14 of Third Level through Fuzzy Method

Table 5.2.3(h) and figure 5.2.3(h) shows the local weights for C14 of Third Level attributes. Consistency Ratio (CR) is 0.0062 which is less than 0.1. This CR value is acceptable to continue Fuzzy AHP analysis. This set of attributes have three attributes including user satisfaction (0.1832), software effectiveness evaluation (0.2239), operational controls (0.5929) and operational controls is highest weighted factor among them.

Table 5.2.3(i): Local Weight of Attributes for C15 of Third Level through Fuzzy Method

	Psychological Acceptability (C151)	**User Satisfaction** (C152)	**Software Effectiveness Evaluation** (C153)	**Operational Controls** (C154)	**Weights**
Psychological Acceptability (C151)	1	1.3651	0.8278	0.3824	**0.1811**
User Satisfaction (C152)	0.7325	1	0.4375	0.2381	**0.1167**
Software Effectiveness Evaluation (C153)	1.2080	2.2857	1	0.8272	**0.2757**
Operational Controls (C154)	2.6151	4.1999	1.2089	1	**0.4265**
					CR=0.0151

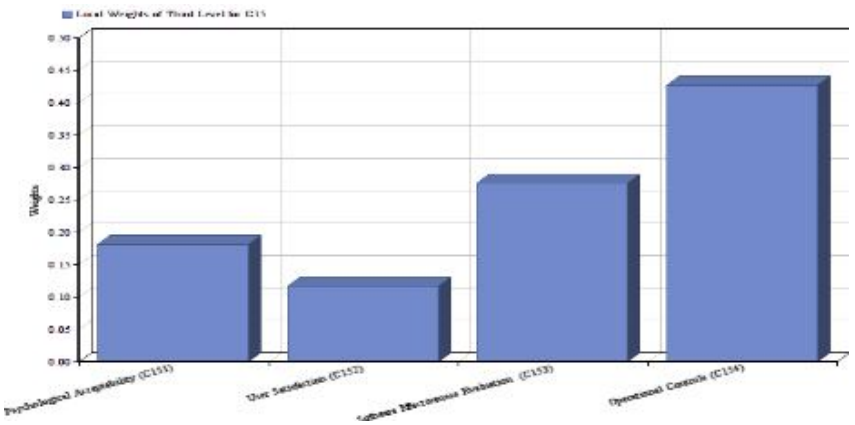

Figure 5.2.3(i): Graphical Representation for C15 of Third Level through Fuzzy Method

Table 5.2.3(i) and figure 5.2.3(i) shows the local weights for C15 of Third Level attributes. Consistency Ratio (CR) is 0.0151 which is less than 0.1. This CR value is acceptable to continue Fuzzy AHP analysis. This set of attributes have four attributes including psychological acceptability (0.1811), user satisfaction (0.1167), software effectiveness evaluation (0.2757), operational controls (0.4265) and operational controls is highest weighted factor among them. Due to repeated attributes in second level, some set of third level attributes are repeated when set of attributes considered as independently. Hence, local weights of third level attributes for C21, C22 and C23 are same as C11, C12 and C13 respectively.

Table 5.2.3(j): Local Weight of Attributes for C25 of Third Level through Fuzzy Method

	Detectability (C251)	**Extensibility** (C252)	**Flexibility** (C253)	**Weights**
Detectability (C251)	1	0.9853	1.3578	**0.3611**
Extensibility (C252)	1.0149	1	1.6269	**0.3873**
Flexibility (C253)	0.7365	0.6147	1	**0.2516**
				C.R.=0.0026

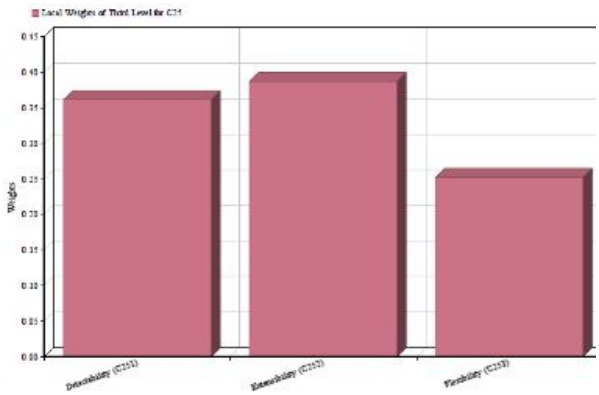

Figure 5.2.3(j): Graphical Representation for C25 of Third Level through Fuzzy Method

Table 5.2.3(j) and figure 5.2.3(j) shows the local weights for C25 of Third Level attributes. Consistency Ratio (CR) is 0.0026 which is less than 0.1. This CR value is acceptable to continue Fuzzy AHP analysis. This set of attributes has three attributes including detectability (0.3611), extensibility (0.3873), flexibility (0.2516) and extensibility is highest weighted factor among them.

Table 5.2.3(k): Local Weight of Attributes for C32 of Third Level through Fuzzy Method

	Psychological Acceptability (C321)	**User Satisfaction** (C322)	**Business Continuity** (C323)	**Operational Controls** (C324)	**Weights**
Psychological Acceptability (C321)	1	1.5973	1.1648	0.7168	**0.2543**
User Satisfaction (C322)	0.6261	1	0.4561	0.3274	**0.1302**
Business Continuity (C323)	0.8585		1	1.0804	**0.2829**
Operational Controls (C324)	1.3951	3.0544	0.9256	1	**0.3326**
					CR=0.0187

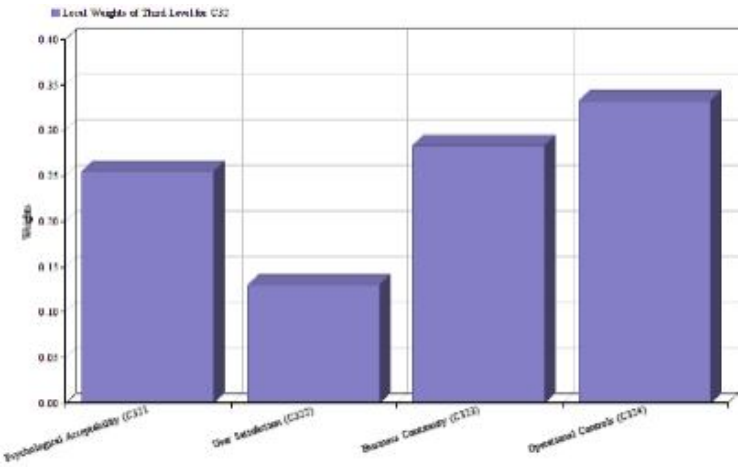

Figure 5.2.3(k): Graphical Representation for C32 of Third Level through Fuzzy Method

Table 5.2.3(k) and figure 5.2.3(k) shows the local weights for C11 of Third Level attributes. Consistency Ratio (CR) is 0.0025 which is less than 0.1. This CR value is acceptable to continue Fuzzy AHP analysis. This set of attributes have four attributes including psychological acceptability (0.2543), user satisfaction (0.1302), business continuity (0.2829), operational controls (0.3326) and operational controls is highest weighted factor among them. Again, local weights of third level attributes for C31, C34 and C35 are same as C12, C14 and C15 respectively. A local weight shows the level wise impact of these attributes and also called independent weights. To evaluate the weights of the security durability attributes throughout the hierarchy, final weights are calculated in next portion.

5.2.4 Final Weights of Each Attribute through Fuzzy Method

Final weights are also called dependent weights of security durability throughout the hierarchy. The final weights (dependent weights) of each attribute through hierarchy are shown in Table 5.2.4(a).

Table 5.2.4(a): The Final Weights of Each Criteria through Hierarchy using Fuzzy Method

The first level	The weight of first level	The second level	Local weight of second level	The final weight of the second level	The third level	The local weight of the third level	The (Global) final weight of the third level
C1	0.4867	C11	0.0946	0.046	C111	0.2400	0.011
					C112	0.0952	0.004
					C113	0.1200	0.006
					C114	0.1032	0.005
					C115	0.4416	0.020
		C12	0.2292	0.112	C121	0.3905	0.044
					C122	0.1694	0.019

					C123	0.2004	0.022
					C124	0.2397	0.027
		C13	0.1192	0.058	C131	0.0733	0.004
					C132	0.0497	0.003
					C133	0.1031	0.006
					C134	0.1271	0.007
					C135	0.1414	0.008
					C136	0.1729	0.010
					C137	0.0760	0.004
					C138	0.2565	0.015
		C14	0.3233	0.157	C141	0.1832	0.029
					C142	0.2239	0.035
					C143	0.5929	0.093
		C15	0.2337	0.114	C151	0.1811	0.021
					C152	0.1167	0.013
					C153	0.2757	0.031
					C154	0.4265	0.049
C2	0.2698	C21	0.1541	0.042	C211	0.2400	0.010
					C212	0.0952	0.004
					C213	0.1200	0.005
					C214	0.1032	0.004
					C215	0.4416	0.018
		C22	0.1692	0.046	C221	0.3905	0.018
					C222	0.1694	0.008
					C223	0.2004	0.009
					C224	0.2397	0.011
		C23	0.1476	0.040	C231	0.0733	0.003
					C232	0.0497	0.002
					C233	0.1031	0.004
					C234	0.1271	0.005
					C235	0.1414	0.006
					C236	0.1729	0.007
					C237	0.0760	0.003
					C238	0.2565	0.010
		C24	0.2214	0.060	C241	-	0.060
		C25	0.3077	0.083	C251	0.3611	0.030
					C252	0.3873	0.032
					C253	0.2516	0.021
C3	0.2435	C31	0.2216	0.054	C311	0.3905	0.021
					C312	0.1694	0.009
					C313	0.2004	0.011
					C314	0.2397	0.013
		C32	0.1596	0.039	C321	0.2543	0.010
					C322	0.1302	0.005
					C323	0.2829	0.011
					C324	0.3326	0.013
		C33	0.1446	0.035	C331	-	0.035
		C34	0.2115	0.052	C341	0.1832	0.009
					C342	0.2239	0.012
					C343	0.5929	0.031
		C35	0.2627	0.064	C351	0.1811	0.012
					C352	0.1167	0.007
					C353	0.2757	0.018
					C354	0.4265	0.027

The hierarchical structure related to security durability attributes is helpful in building the effective security design of software. The decomposition of security durability attributes has been considered in three levels viz., level 1, level 2 and level 3. Based on the results, rank of each attributes is obtained at level 1, 2 and 3.

On the basis of final weightages, evaluation of the ranks of each attribute for improving security durability/security life span of software is illustrated. The required security durability attributes are extracted from figure 5.2 (a) and table 5.2.4(a) shows the importance of each attribute throughout the hierarchy in the form of priorities. Repeated attributes of level 2 and level 3 are removed and figure 5.2.4 (a) and figure 5.2.4 (b) shows the final priorities of security durability attributes at level 2 and level 3.

Second Level Characteristics	The final weight of the second level	Final Ranks of the Second Level
Availability	0.046	10
Reliability	0.112	3
Maintainability	0.058	7
Confidentiality	0.157	1
Authentication	0.114	2
~~Availability~~	~~0.042~~	~~12~~
~~Reliability~~	~~0.046~~	~~11~~
~~Maintainability~~	~~0.040~~	~~13~~
Accountability	0.060	6
Survivability	0.083	4
~~Reliability~~	~~0.054~~	~~8~~
Consumer Integrity	0.039	14
~~Accountability~~	~~0.035~~	~~15~~
~~Confidentiality~~	~~0.052~~	~~9~~
~~Authentication~~	~~0.064~~	~~5~~

Set of Attributes without Repetition

Priority	Characteristics of Level 2
1	Confidentiality
2	Authentication
3	Reliability
4	Survivability
5	Accountability
6	Maintainability
7	Availability
8	Consumer Integrity

Figure 5.2.4(a): Second level Attributes without Repetition

Third Level Characteristics	The Final Weight of the Third Level	Final Ranks of the Third Level
~~Auditability~~	~~0.011~~	~~29~~
~~Feasibility~~	~~0.004~~	~~52~~
~~Accessibility~~	~~0.006~~	~~45~~
~~Software Effective Evaluation~~	~~0.005~~	~~48~~
~~Operational Controls~~	~~0.020~~	~~18~~
Feasibility	0.044	4
Time Efficiency	0.019	19
~~User Satisfaction~~	~~0.022~~	~~14~~
Business Continuity	0.027	12
~~Auditability~~	~~0.004~~	~~53~~
Scalability	0.003	58
Traceability	0.006	46
Detectability	0.007	42
~~Extensibility~~	~~0.008~~	~~40~~
~~Flexibility~~	~~0.010~~	~~35~~
~~Accessibility~~	~~0.004~~	~~54~~
~~Time-Efficiency~~	~~0.015~~	~~23~~
User Satisfaction	0.029	11
~~Software Effective Evaluation~~	~~0.057~~	~~5~~
Operational Controls	0.093	1
Psychological Acceptability	0.021	15
~~User Satisfaction~~	~~0.013~~	~~24~~
~~Software Effective Evaluation~~	~~0.031~~	~~8~~
~~Operational Controls~~	~~0.049~~	~~3~~
~~Auditability~~	~~0.010~~	~~34~~
~~Feasibility~~	~~0.004~~	~~55~~
~~Accessibility~~	~~0.005~~	~~49~~
~~Software Effective Evaluation~~	~~0.004~~	~~56~~
~~Operational Controls~~	~~0.018~~	~~20~~
~~Feasibility~~	~~0.014~~	~~23~~
~~Time-Efficiency~~	~~0.008~~	~~41~~
~~User Satisfaction~~	~~0.009~~	~~37~~
~~Business Continuity~~	~~0.011~~	~~30~~
~~Auditability~~	~~0.003~~	~~59~~
~~Scalability~~	~~0.002~~	~~61~~
~~Traceability~~	~~0.004~~	~~57~~
~~Detectability~~	~~0.005~~	~~50~~
~~Extensibility~~	~~0.006~~	~~47~~
~~Flexibility~~	~~0.007~~	~~43~~
~~Accessibility~~	~~0.003~~	~~60~~
~~Time-Efficiency~~	~~0.015~~	~~25~~
Software Effective Evaluation	0.060	2
Detectability	0.030	10
Extensibility	0.032	7
Flexibility	0.021	16
~~Feasibility~~	~~0.021~~	~~17~~
~~Time-Efficiency~~	~~0.009~~	~~38~~
User Satisfaction	0.011	31
~~Business Continuity~~	~~0.013~~	~~26~~
~~Psychological Acceptability~~	~~0.010~~	~~36~~
~~User Satisfaction~~	~~0.005~~	~~51~~
~~Business Continuity~~	~~0.011~~	~~32~~
~~Operational Controls~~	~~0.013~~	~~28~~
~~Software Effective Evaluation~~	~~0.033~~	~~6~~
~~User Satisfaction~~	~~0.008~~	~~39~~
~~Software Effective Evaluation~~	~~0.012~~	~~27~~
~~Operational Controls~~	~~0.033~~	~~9~~
~~Psychological Acceptability~~	~~0.012~~	~~28~~
~~User Satisfaction~~	~~0.007~~	~~44~~
~~Software Effective Evaluation~~	~~0.018~~	~~22~~
~~Operational Controls~~	~~0.027~~	~~13~~

Set of Attributes without Repetition

Priority	Characteristics of Level 3
1	Operational Controls
2	Software Effectiveness Evaluation
3	Feasibility
4	User Satisfaction
5	Time-efficiency
6	Auditability
7	Psychological Acceptability
8	Business Continuity
9	Accessibility
10	Extensibility
11	Flexibility
12	Detectability
13	Scalability
14	Traceability

Figure 5.2.4(b): Third level attributes without repetition

Figure 5.2.4 (a) and figure 5.2.4 (b) shows the final priorities of security durability attributes at level 2 and level 3 after removing the repeated attributes. These priorities will help towards creating the development suggestions/guidelines.

5.3 Procedure for Improving Security Durability of Software

The aim of the proposed work is not limited to quantify security durability but to produce guidelines on the basis of that quantification. The guidelines inferred from the quantification will surely help the developers to improve security durability of software during its development. To produce any guidelines for developers related to design, it is important to consider properties of design. Object oriented design properties are measured using its corresponding security metrics [103]. Further, object-oriented security metrics are useless if they are not mapped to security durability parameters. There are numerous security metric suites available to predict security of the software namely Vulnerable Association of an Object Oriented Design(VA_OOD) [115], Security Requirements Statistics (SRs) [116], Number of Design stage Security Errors (NDSE) [117], Critical Class Coupling (CCC) [118], Critical Class Extensibility(CCE) [119], Critical Super Class Propagation(CSP) [120], Classified Method Inheritance (CMI) [121], Classified Attributes Inheritance(CAI) [122], Critical Design Propagation (CDP) [123], Classified Instance Data Accessibility(CIDA) [124], Classified Methods Weight (CMW) and many more [125]. The names specified above here are security metrics for design phase. These metrics are specifically used for measuring the impact of the properties. For example, to measure coupling of classes, Critical Class Coupling (CCC) is used by most of the practitioners [125].

Most of the design properties have positive impacts on security attributes including service-oriented design and object-oriented design etc. [19]. On the other hand, each design strategy has its own positive and negative impacts on security services of software. In this work, researcher suggests only eight security metrics to developers that may be helpful for measuring and achieving the priorities of third level factors including Critical Class Coupling (CCC), Critical Class Extensibility (CCE), Critical Super Class Propagation (CSP), Classified Method Inheritance (CMI), Classified Attributes Inheritance (CAI), Critical Design Propagation (CDP), Classified Instance Data Accessibility (CIDA) and Classified Methods Weight (CMW). Through the impact of third level priorities, second level, first level and overall security durability are measured and achieved. Security durability attributes (third level) affect many

design attributes and impact of these attributes may be helpful for assessment through suggested security metrics as:

- Auditability affects design properties such as reusability [126], discoverability [127], design by contract [128] and design size [129]. With the help of CMI and CAI metrics, affected design properties of auditability may be measured and improved [124]. Further, CMI measures the ratio between number of classified methods and total number of classified methods and CAI measures the ratio between numbers of classified attributes and total number of classified attributes.

- Scalability affects design properties such as coupling [130] and reusability [126]. With the help of CCC and CMI metrics, affected design properties of scalability may be measured and improved [125]. Further, CCC helps to measure the ratio between the numbers of all classes linked with classified attributes.

- Feasibility affects design properties such as reusability [126] and discoverability [127]. With the help of CAI and CMI metrics, affected design properties of feasibility may be measured and improved.

- Traceability affects design properties such as coupling [130], abstraction [126] and discoverability [127]. With the help of CCC and CSP metrics, affected design properties of traceability may be measured and improved [126]. Further, CSP helps to measure the ratio between the numbers of critical super classes and total number of critical classes in an inheritance hierarchy; and also helps to implement the abstraction.

- Detectability affects design properties such as autonomy [125], discoverability [127] and cohesion [130]. With the help of CCE metric, affected design properties of detectability may be measured and improved [125]. Further, CCE helps to measure the ratio between numbers of non-finalized classes in a design with the critical classes in that design.

- Accessibility affects design properties such as complexity [19] and design size [129]. With the help of CDP and CIDA metrics, affected design properties of accessibility may be measured and improved [125]. Further, CDP measures the ratio between the number of critical classes and total number of classes in a design; and measure the impact of the size of a certain design on security. CIDA is helpful to measure the ratio between the number of

classified instance public attributes and total number of classified attributes in a class; and it also measures the impact of the size of a certain design on security.

- Time-efficiency affects design properties such as design size [129] and reusability [126]. With the help of CMI and CAI metrics, affected design properties of time-efficiency may be measured and improved.

- Extensibility affects design properties such as complexity [19] and reusability [126]. With the help of CMI and CAI metrics, affected design properties of extensibility may be measured and improved.

- Psychological acceptability affects design properties such as abstraction [125], design by contract [128] and cohesion [130]. With the help of CSP metric, affected design properties of psychological acceptability may be measured and improved.

- User satisfaction affects design properties such as abstraction [126] and autonomy [125]. With the help of CSP and CCE metrics, affected design properties of user satisfaction may be measured and improved.

- Software effectiveness evaluation affects design properties such as abstraction [126] and coupling [130]. With the help of CCE, CMI, CAI and CSP metrics, affected design properties of software effectiveness evaluation may be measured and improved.

- Business continuity affects design properties such as coupling and cohesion [130]. With the help of CCC and CMW metrics, affected design properties of business continuity may be measured and improved.

- Flexibility affects design properties such as coupling [130] and statelessness [131]. With the help of CMW, CDP, and CCC metrics, affected design properties of flexibility may be measured and improved [125]. Further, CMW helps to measure the ratio between the numbers of classified methods and total number of methods in a given class. CDP measures the ratio between the number of critical classes and total number of classes; and also helps to measures the impact of the size of a certain design on security.

- Also, operational controls affect design properties such as coupling [130] and statelessness [131]. With the help of CMW, CDP, and CCC metrics, affected design properties of operational controls may be measured and improved.

Through the measurement of third level attributes, the impact of second level attributes of security durability may be measured. Further, to measure and improve the impact of second level attributes, following are the referrals.

- Confidentiality is affected by third level attributes including user satisfaction, software effective evaluation, and operational controls. With the help of the metrics of design properties for these attributes, the impact of confidentiality may be measured and improved.
- Authentication is affected by third level attributes including psychological acceptability, user satisfaction, software effectiveness evaluation and operational controls. With the help of the metrics of design properties for these attributes, the impact of authentication may be measured and improved.
- Reliability is affected by third level attributes including feasibility, time-efficiency, user satisfaction and business continuity. With the help of the metrics of design properties for these attributes, the impact of reliability may be measured and improved.
- Survivability is affected by third level attributes including detectability, extensibility and flexibility. With the help of the metrics of design properties for these attributes, the impact of survivability may be measured and improved.

Through the measurement of second level attributes, the impact of first level attributes of security durability may be measured. Further, to measure and improve the impact of first level attributes, following are the referrals.

- Dependability is affected by second level attributes including availability, reliability, maintainability, confidentiality and authentication. With the help of the impact of these second level attributes, the impact of dependability may be measured and improved.
- Trustworthiness is affected by second level attributes including availability, reliability, maintainability, accountability and survivability. With the help of the impact of these second level attributes, the impact of trustworthiness may be measured and improved.

- Human trust is affected by second level attributes including reliability, consumer integrity, accountability, confidentiality and authentication. With the help of the impact of these second level attributes, the impact of human trust may be measured and improved.

- With the help of given final priorities of level 1, 2 and 3 and above discussion, developers should focus to enhance the high prioritized attributes. Measurement through the metrics is necessary for enhancing the impact of these attributes on overall security durability of software services. To create the suggestions/ guidelines, procedure is given in chapter 4. Further, recommendations to better implementation and improvement are descriptively given below:

- Improve security durability awareness among developers by adequate education and training to achieve sound security durability culture in organizational environment during use of software services.

- Economic aspect of security life span should be clearly understood and addressed as one of important factors for organization in recent information era.

- Periodically review the performance of security durability policy implementations using the MCDM techniques because these techniques hail from academia as well as software industry so as to realize the real-world practices.

- The development guidelines that have positive effect on the highest priority security durability attribute, which in this case, dependability, are gathered.

- On the basis of assessment, security metric for dependability is prepared and calculated

- At focus, dependability, human trust and trustworthiness are the important factors for a security durability of software services.

- Importance of level 1, level 2 and level 3 attributes are shown in figure 5.2.4 (a) and figure 5.2.4 (b) and must be followed by developers.

- If any dataset does not include these high priority attributes than developers must include those through the metrics to ensure security durability.

- In level 1, dependability, trustworthiness, and human trust respectively are important attributes.
- In level 2, confidentiality, authentication and reliability are more desirable attributes and necessary attribute among all other attributes of security durability which are shown in figure 5.2.4 (a).
- In level 3, operational controls, software effectiveness evaluation and feasibility are more essential attributes and required attribute among all other attributes of security durability which are shown in figure 5.2.4 (b).
- This assessment does not have overlapping, duplicate and conflicting relationships but if confliction occurs than developers should take the high prioritized ones.

To analyze the impact of given priorities, suggestions and recommendations, researcher evaluated the performance of security durability in both subjective and objective perspectives. Further, subjective assessment is done in previous portion of this chapter. To evaluate the objective assessment, this work is taking two alternatives of BBAU software say, version 1 and version 2. The process is discussed in the next portion.

5.4 Ratings of Attributes through Fuzzy Method

A rating is the evaluation of something, in terms of quality, quantity, or some combination of both. According to oxford dictionary “Rating is a classification of something based on a comparative assessment of their quality, standard, or performance” [132].

To evaluate the objective weightages, researcher has taken the ratings of security durability attributes from the development team for BBAU software including version 1 and version 2. Old design of the software is called version 1 and modified design of the software is called version 2. Due to sensitivity of the software, development organizations have not given the design structure to the researcher but two teams of the development organizations helped for the experiments which are called team 1 and team 2. Team 1 helped to reform the old design into modified design and team 2 helped to give the ratings of security durability attributes.

Further, according to given priorities and recommendations, the suggested metrics will be helpful for team 1 to modify the design. The suggested metrics may be helpful for team 1 to achieve the resulted priorities and reform the security design of software. To measure the

impact of security durability attributes for version 1 and version 2, team 2 collected the ratings. Researcher took the ratings and converted the linguistic values into numerical values with the help of rating scale table 3.6.2(b) and fuzzy aggregation method is used to evaluate the ratings (also called objective weightages) of security durability attributes for version 1 and version 2. Further, fuzzy aggregation method is used in various research areas for decision making, rating and so on [97, 134]. To fuzzify and aggregate the ratings next portion discusses the mechanism and implementation.

5.4.1 Fuzzified Average Ratings

Ratings of security durability attributes are collected at level 1, level 2 and level 3. With the help of rating scale table 3.6.2(b), linguistic values converted into numerical values and numerical values into Triangular Fuzzy Numbers (TFN). To confine the vagueness of the parameters which are related to alternatives including TFN is used [102]. With the help of equations (1, 3-5), fuzzified average ratings are evaluated. Table 5.4.1(a) is shown the fuzzified average ratings of security durability attributes for version 1 and version 2.

Table 5.4.1(a): Fuzzified Average Ratings

S. No.	Characteristics of Level 1	Old Version (Version 1)	Modified Version (Version 2)
		Fuzzified Average Rating	
1	Dependability	0.445, 0.615, 0.755	0.59, 0.79, 0.95
2	Trustworthiness	0.455, 0.64, 0.74	0.64, 0.84, 0.97
3	Human Trust	0.44, 0.60, 0.74	0.62, 0.82, 0.96
S. No	Characteristics of Level 2	Fuzzified Average Rating	
1	Reliability	0.53, 0.72, 0.865	0.62, 0.81, 0.94
2	Availability	0.46, 0.63, 0.775	0.63, 0.82, 0.94
3	Authentication	0.38, 0.55, 0.71	0.67, 0.85, 0.95
4	Maintainability	0.445, 0.635, 0.79	0.65, 0.84, 0.95
5	Confidentiality	0.56, 0.72, 0.835	0.51, 0.70, 0.86
6	Accountability	0.445, 0.615, 0.765	0.64, 0.83, 0.95
7	Consumer Integrity	0.46, 0.635, 0.78	0.73, 0.90, 0.99
8	Survivability	0.495, 0.68, 0.83	0.69, 0.87, 0.98
S. No.	Characteristics of Level 3	Fuzzified Average Rating	
1	Software Effectiveness Evaluation	0.66, 0.60, 0.875	0.61, 0.75, 0.93
2	User Satisfaction	0.64, 0.81, 0.935	0.52, 0.64, 0.84
3	Feasibility	0.49, 0.57, 0.835	0.53, 0.65, 0.89
4	Operational Controls	0.75, 0.67, 0.985	0.66, 0.78, 0.97
5	Time-efficiency	0.35, 0.52, 0.77	0.69, 0.85, 0.99
6	Auditability	0.56, 0.6, 0.875	0.47, 0.58, 0.83
7	Psychological Acceptability	0.43, 0.58, 0.90	0.61, 0.72, 0.96
8	Business Continuity	0.42, 0.57, 0.905	0.52, 0.57, 0.90
9	Accessibility	0.49, 0.61, 0.795	0.50, 0.61, 0.84

10	Extensibility	0.44, 0.60, 0.89	0.46, 0.56, 0.82
11	Flexibility	0.50, 0.66, 0.84	0.43, 0.54, 0.79
12	Detectability	0.51, 0.56, 0.83	0.49, 0.59, 0.85
13	Scalability	0.46, 0.62, 0.895	0.51, 0.66, 0.85
14	Traceability	0.40, 0.57, 0.845	0.49, 0.57, 0.87

Table 5.4.1(a) shows the fuzzified average ratings of security durability attributes (attributes of level 1, level 2 and level 3) for version 1 and version 2. Local ratings of security durability attributes for version 1 and version 2 is evaluated in next portion.

5.4.2 Defuzzification and Local Ratings

With the help of equations (7-9) (discussed in chapter 3), local ratings of security durability attributes are estimated. These local ratings are also called independent ratings. Further, table 5.4.2 (a) is showing the local ratings for version 1 and version 2.

Table 5.4.2(a): Local Rating of the Attributes for Level 1, 2 and 3 through Fuzzy Method

S. No.	Characteristics of Level 1	Old Version (Version 1)	Modified Version (Version 2)
		Defuzzified Local Rating	
1	Dependability	0.608	0.78
2	Trustworthiness	0.619	0.82
3	Human Trust	0.595	0.81
S. No.	**Characteristics of Level 2**	**Defuzzified Local Rating**	
1	Reliability	0.709	0.79
2	Availability	0.624	0.80
3	Authentication	0.548	0.83
4	Maintainability	0.626	0.82
5	Confidentiality	0.709	0.69
6	Accountability	0.610	0.81
7	Consumer Integrity	0.628	0.88
8	Survivability	0.671	0.85
S. No.	**Characteristics of Level 3**	**Defuzzified Local Rating**	
1	Software Effectiveness Evaluation	0.626	0.76
2	User Satisfaction	0.799	0.66
3	Feasibility	0.616	0.68
4	Operational Controls	0.769	0.79
5	Time-efficiency	0.540	0.84
6	Auditability	0.659	0.61
7	Psychological Acceptability	0.623	0.75
8	Business Continuity	0.616	0.64
9	Accessibility	0.626	0.64
10	Extensibility	0.633	0.60
11	Flexibility	0.665	0.58
12	Detectability	0.615	0.63

13	Scalability	0.649	0.67
14	Traceability	0.596	0.62

Table 5.4.2(a) shows the local ratings of security durability attributes for level 1, level 2 and level 3 respectively. Ratings for the set of first level attributes have three attributes including dependability (0.608, 0.780), trustworthiness (0.619, 0.820) and human trust (0.595, 0.810) for version 1 and version 2 and trustworthiness is highest rated factor among them for version 1 and version 2 respectively. Ratings for the set of second level attributes have eight attributes including reliability (0.709, 0.790), availability (0.624, 0.800), authentication (0.548, 0.830), maintainability (0.626, 0.820), confidentiality (0.709, 0.690), accountability (0.610, 0.810), consumer integrity (0.628, 0.880), survivability (0.671, 0.850) for version 1 and version 2 respectively. Reliability and confidentiality are equally highest rated factors among all for version 1. Consumer integrity is highest rated factor among all for version 2.

Ratings for the set of third level attributes have fourteen attributes including software effectiveness evaluation (0.626, 0.760), user satisfaction (0.799, 0.660), feasibility (0.616, 0.680), operational controls (0.769, 0.790), time-efficiency (0.540, 0.840), auditability (0.659, 0.610), psychological acceptability (0.623, 0.750), business continuity (0.616, 0.640), accessibility (0.626, 0.640), extensibility (0.633, 0.600), flexibility (0.665, 0.580), detectability (0.615, 0.630), scalability (0.649, 0.670), traceability (0.596, 0.620). User satisfaction is highest rated factor among all for version 1. Time-efficiency is highest rated factor among all for version 2. Further, local ratings are showing the level wise impact of these attributes for version 1 and version 2 and also called independent ratings. To evaluate the impact of the security durability attributes throughout the hierarchy, final ratings are calculated in next portion.

5.4.3 Final Rating of Each Attribute through Fuzzy Method

Table 5.4.2(a) above shows the independent ratings of every attribute at level 1, 2 and 3. Next step in this row is to calculate the final ratings of attributes according to their place in hierarchy. For calculating the final ratings, the lower level ratings are multiplied to the higher level ratings. Table 5.4.3(a) shows the final ratings of each attribute through fuzzy method.

Table 5.4.3(a): Final Ratings of Each Attribute through Fuzzy Method

The first level	The Ratings of durability factors of the first level		The second level	Local Ratings of second level		The final Ratings of the second level		The level of the third level	The local Ratings of the third level		The final Ratings of the third level	
	Version 1	Version 2		Version 1	Version 2	Version 1	Version 2		Version 1	Version 2	Version 1	Version 2
C1	0.608	0.78	C11	0.624	0.8	0.379	0.624	C111	0.659	0.760	0.250	0.474
								C112	0.616	0.660	0.234	0.412
								C113	0.626	0.680	0.237	0.424
								C114	0.781	0.790	0.296	0.493
								C115	0.769	0.840	0.292	0.524
			C12	0.709	0.79	0.431	0.616	C121	0.616	0.660	0.266	0.407
								C122	0.540	0.610	0.233	0.376
								C123	0.799	0.750	0.344	0.462
								C124	0.616	0.640	0.266	0.394
			C13	0.626	0.82	0.381	0.640	C131	0.659	0.760	0.251	0.486
								C132	0.649	0.640	0.247	0.409
								C133	0.596	0.600	0.227	0.384
								C134	0.615	0.580	0.234	0.371
								C135	0.633	0.630	0.241	0.403
								C136	0.665	0.670	0.253	0.429
								C137	0.626	0.680	0.238	0.435
								C138	0.540	0.610	0.206	0.390
			C14	0.709	0.69	0.431	0.538	C141	0.799	0.750	0.344	0.404
								C142	0.781	0.790	0.337	0.425
								C143	0.769	0.870	0.331	0.468
			C15	0.578	0.83	0.351	0.647	C151	0.623	0.620	0.219	0.401
								C152	0.799	0.750	0.281	0.486
								C153	0.781	0.790	0.274	0.511
								C154	0.769	0.840	0.270	0.544
C2	0.619	0.82	C21	0.624	0.8	0.386	0.656	C211	0.659	0.760	0.254	0.499
								C212	0.616	0.660	0.238	0.433
								C213	0.626	0.680	0.242	0.446
								C214	0.781	0.790	0.302	0.518
								C215	0.769	0.840	0.297	0.551
			C22	0.709	0.79	0.439	0.648	C221	0.616	0.660	0.270	0.428
								C222	0.540	0.610	0.237	0.395
								C223	0.799	0.750	0.351	0.486
								C224	0.616	0.640	0.270	0.415
			C23	0.626	0.82	0.387	0.672	C231	0.659	0.760	0.255	0.511
								C232	0.649	0.640	0.251	0.430
								C233	0.596	0.600	0.231	0.403
								C234	0.615	0.580	0.238	0.390
								C235	0.633	0.630	0.245	0.424
								C236	0.665	0.670	0.258	0.451
								C237	0.626	0.680	0.243	0.457
								C238	0.540	0.610	0.209	0.410
			C24	0.61	0.81	0.483	0.648	C241	0.781	0.790	0.378	0.512
			C25	0.671	0.85	0.415	0.697	C251	0.615	0.580	0.255	0.404
								C252	0.633	0.630	0.263	0.439
								C253	0.665	0.670	0.276	0.467
C3	0.595	0.87	C31	0.709	0.79	0.422	0.687	C311	0.616	0.660	0.260	0.454
								C312	0.540	0.610	0.228	0.419
								C313	0.799	0.750	0.337	0.515
								C314	0.616	0.640	0.260	0.440
			C32	0.628	0.88	0.374	0.766	C321	0.623	0.620	0.233	0.475
								C322	0.799	0.750	0.299	0.574
								C323	0.781	0.640	0.292	0.490
								C324	0.769	0.840	0.287	0.643
			C33	0.61	0.81	0.363	0.705	C331	0.781	0.790	0.283	0.557
			C34	0.709	0.69	0.422	0.600	C341	0.799	0.750	0.337	0.450
								C342	0.781	0.790	0.329	0.474

								C343	0.769	0.840	0.324	0.504
			C35	0.548	0.83	0.326	0.722	C351	0.623	0.620	0.203	0.448
								C352	0.799	0.750	0.261	0.542
								C353	0.781	0.790	0.255	0.570
								C354	0.769	0.840	0.251	0.607

Many attributes at level 2 and level 3 are repeated but their impact on its higher level attributes is different. With the help of hierarchy, dependent ratings are evaluated but there are different impacts of same attribute. With the help of final ratings and weights, security durability of software is estimated for version 1 and version 2 in next portion.

5.5 Assessment of Security Durability through Fuzzy Method

From equation (16), security durability is assessed for two alternatives i.e. version 1 and version 2 with the help of final ratings (R_i) and weights (W_i) of attributes. The calculation of the assessment as follows:

$$[\,0.011\ 0.004\ 0.006\ 0.005\ 0.020\ 0.044\ 0.019\ 0.022\ 0.027\ 0.004\ 0.003\ 0.006\ 0.007\ 0.008\ 0.010\ 0.004\ 0.015\ 0.029\ 0.035\ 0.093\ 0.021\ 0.013\ 0.031\ 0.049\ 0.010\ 0.004\ 0.005\ 0.004\ 0.018\ 0.018\ 0.008\ 0.009\ 0.011\ 0.003\ 0.002\ 0.004\ 0.005\ 0.006\ 0.007\ 0.003\ 0.010\ 0.060\ 0.030\ 0.032\ 0.021\ 0.021\ 0.009\ 0.011\ 0.013\ 0.010\ 0.005\ 0.011\ 0.013\ 0.035\ 0.009\ 0.012\ 0.031\ 0.012\ 0.007\ 0.018\ 0.027\,] \times$$

$$\begin{bmatrix} 0.250 & 0.474 \\ 0.234 & 0.412 \\ 0.237 & 0.424 \\ 0.296 & 0.493 \\ 0.292 & 0.524 \\ 0.266 & 0.407 \\ 0.233 & 0.376 \\ 0.344 & 0.462 \\ 0.266 & 0.394 \\ 0.251 & 0.486 \\ 0.247 & 0.409 \\ 0.227 & 0.384 \\ 0.234 & 0.371 \\ 0.241 & 0.403 \\ 0.253 & 0.429 \\ 0.238 & 0.435 \\ 0.206 & 0.390 \\ 0.344 & 0.404 \\ 0.337 & 0.425 \\ 0.331 & 0.468 \\ 0.219 & 0.401 \\ 0.281 & 0.486 \\ 0.274 & 0.511 \\ 0.270 & 0.544 \\ 0.254 & 0.499 \\ 0.238 & 0.433 \\ 0.242 & 0.446 \\ 0.302 & 0.518 \\ 0.297 & 0.551 \\ 0.270 & 0.428 \\ 0.237 & 0.395 \\ 0.351 & 0.486 \\ 0.270 & 0.415 \\ 0.255 & 0.511 \\ 0.251 & 0.430 \\ 0.231 & 0.403 \\ 0.238 & 0.390 \\ 0.245 & 0.424 \\ 0.258 & 0.451 \\ 0.243 & 0.457 \\ 0.209 & 0.410 \\ 0.378 & 0.512 \\ 0.255 & 0.404 \\ 0.263 & 0.439 \\ 0.276 & 0.467 \\ 0.260 & 0.454 \\ 0.228 & 0.419 \\ 0.337 & 0.515 \\ 0.260 & 0.440 \\ 0.233 & 0.475 \\ 0.299 & 0.574 \\ 0.292 & 0.490 \\ 0.287 & 0.643 \\ 0.283 & 0.557 \\ 0.337 & 0.450 \\ 0.329 & 0.474 \\ 0.324 & 0.504 \\ 0.203 & 0.448 \\ 0.261 & 0.542 \\ 0.255 & 0.570 \\ 0.251 & 0.607 \end{bmatrix} = \begin{matrix} \text{Version 1} \\ \text{Version 2} \end{matrix} = \begin{bmatrix} 0.2852 \\ 0.4700 \end{bmatrix}$$

Where, values of final weights are shown in the row and values of final ratings are shown in the columns. This work is done for the two alternatives i.e. version 1 and version 2. Hence, final ratings are shown in the two columns. Overall security durability is shown in table 5.5 (a).

Table 5.5(a): Overall Security Durability through Fuzzy Method

Security Durability		
	Version 1	**Version 2**
Security Durability	0.2852	0.4700

Figure 5.5(a): Graphical representation of Overall Security Durability through Fuzzy Method

Table 5.5(a) and figure 5.5(a) are showing the values of security durability of BBAU software. Value of security durability for old version (version 1) is 0.2852 and value of security durability for modified version (version 2) is 0.4700. Again, with the help of final weights, final ratings of both version and equation 16, impact of security durability at first level is calculated which is shown in table 5.5 (b).

Table 5.5(b): Security Durability Impact at Level 1 through Fuzzy Method

Contribution of Security Durability at Level 1			
S. No.	**Characteristics of Level 1**	**Version 1**	**Version 2**
1	**Dependability**	0.1391	0.2187
2	**Trustworthiness**	0.0782	0.1246
3	**Human Trust**	0.0679	0.1267

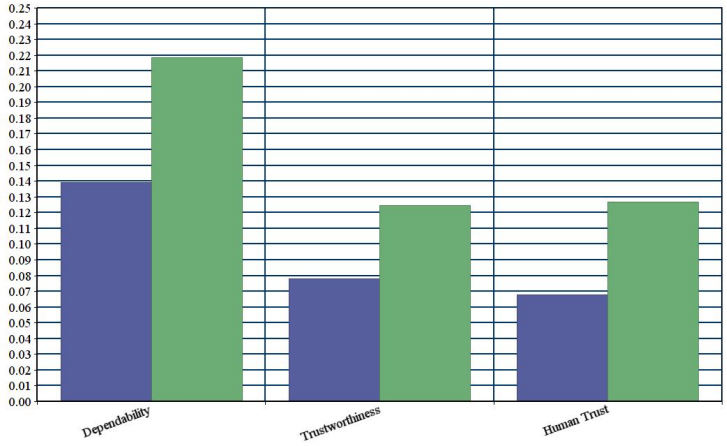

Figure 5.5(b): Graphical representation of Security Durability Impact at Level 1 through Fuzzy Method

Table 5.5(b) and figure 5.5(b) are showing the values of security durability on first level attributes. Contributions of security durability for dependability are 0.1391 and 0.2187 for version 1 and version 2 respectively. Contributions of security durability for trustworthiness are 0.0782 and 0.1246 for version 1 and version 2 respectively. Contributions of security durability for human trust are 0.0679 and 0.1267 for version 1 and version 2 respectively. Again, with the help of final weights, final ratings of both version and equation 16, impact of security durability at second level is calculated which is shown in table 5.5 (c).

Table 5.5(c): Security Durability Impact at Level 2 through Fuzzy Method

Contribution of Security Durability at Level 2			
S. No.	**Characteristics of Level 2**	**Version 1**	**Version 2**
1	**Reliability**	0.0584	0.0903
2	**Availability**	0.0237	0.0433
3	**Authentication**	0.0456	0.0931
4	**Maintainability**	0.0227	0.0403
5	**Confidentiality**	0.0696	0.0955
6	**Accountability**	0.0326	0.0502
7	**Consumer Integrity**	0.0108	0.0214
8	**Survivability**	0.0219	0.0360

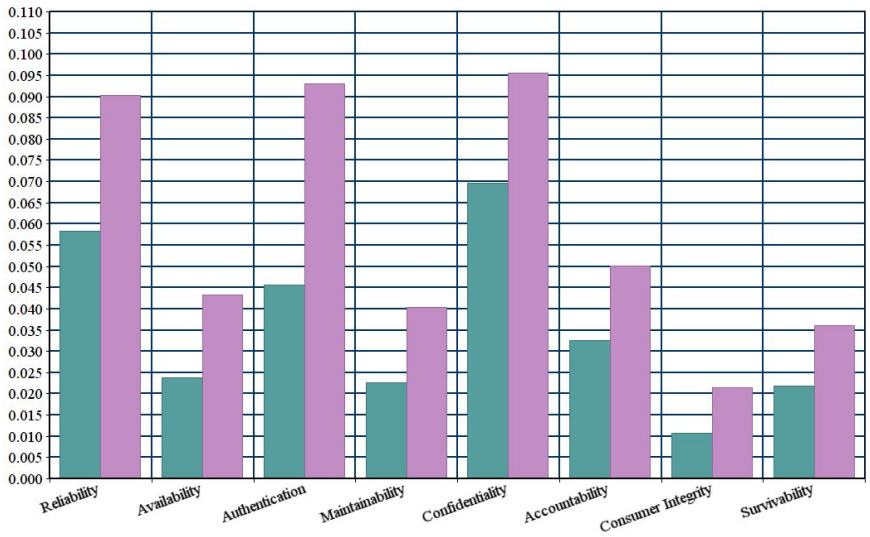

Figure 5.5(c): Graphical representation of Security Durability Impact at Level 2 through Fuzzy Method

Table 5.5(c) and figure 5.5(c) are showing the values of security durability on second level attributes. Contributions of security durability for reliability are 0.0584 and 0.0903 for version 1 and version 2 respectively. Contributions of security durability for availability are 0.0237 and 0.0433 for version 1 and version 2 respectively. Contributions of security durability for authentication are 0.0456 and 0.0931 for version 1 and version 2 respectively. Contributions of security durability for maintainability are 0.0227 and 0.0403 for version 1 and version 2 respectively. Contributions of security durability for confidentiality are 0.0696 and 0.0955 for version 1 and version 2 respectively. Contributions of security durability for accountability are 0.0326 and 0.0502 for version 1 and version 2 respectively. Contributions of security durability for consumer integrity are 0.0108 and 0.0214 for version 1 and version 2 respectively. Contributions of security durability for survivability are 0.0219 and 0.0360 for version 1 and version 2 respectively. Again, with the help of final weights, final ratings of both version and equation 16, impact of security durability at third level is calculated which is shown in table 5.5 (d).

Table 5.5(d): Security Durability Impact at Level 3 through Fuzzy Method

Contribution of Security Durability at Level 3			
S. No.	**Characteristics of Level 3**	**Version 1**	**Version 2**
1	**Software Effectiveness Evaluation**	0.0641	0.1014
2	**User Satisfaction**	0.0344	0.0490
3	Feasibility	0.0239	0.0385
4	**Operational Controls**	0.0758	0.1310
5	**Time-efficiency**	0.0136	0.0240
6	**Auditability**	0.0071	0.0137

7	**Psychological Acceptability**	0.0094	0.0185
8	**Business Continuity**	0.0167	0.0263
9	**Accessibility**	0.0043	0.0079
10	**Extensibility**	0.0118	0.0198
11	**Flexibility**	0.0101	0.0173
12	**Detectability**	0.0105	0.0167
13	**Scalability**	0.0012	0.0021
14	**Traceability**	0.0023	0.0039

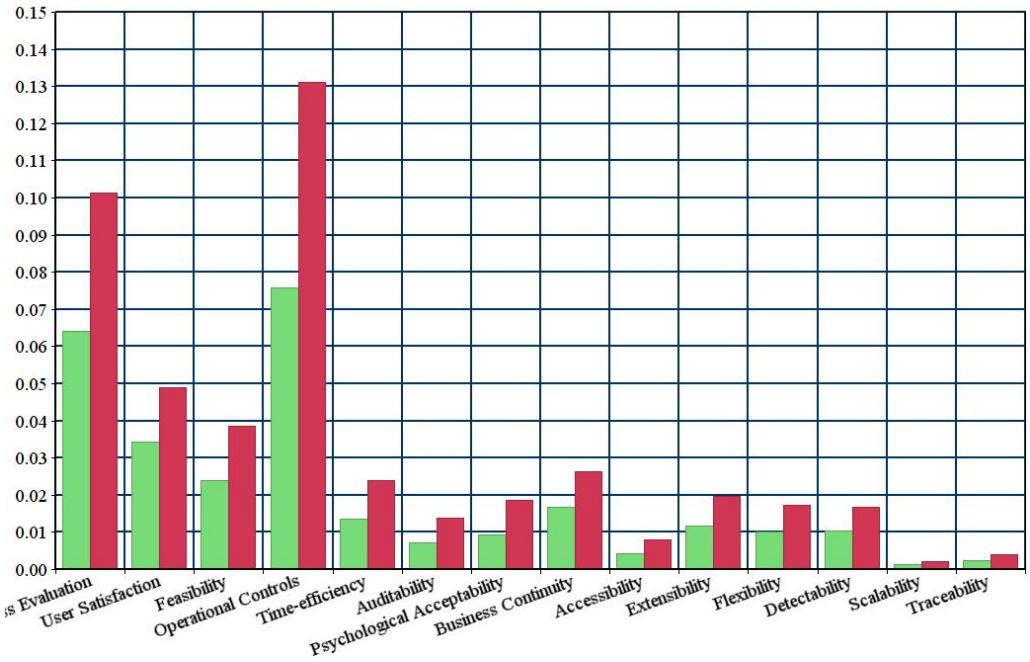

Figure 5.5(d): Graphical representation of Security Durability Impact at Level 3 through Fuzzy Method

Table 5.5(d) and Figure 5.5(d) are showing the values of security durability on second level attributes. Contributions of security durability for software effectiveness evaluation are 0.0641 and 0.1014 for version 1 and version 2 respectively. Contributions of security durability for user satisfaction are 0.0344 and 0.0490 for version 1 and version 2 respectively. Contributions of security durability for feasibility are 0.0239 and 0.0385 for version 1 and version 2 respectively. Contributions of security durability for operational controls are 0.0758 and 0.1310 for version 1 and version 2 respectively. Contributions of security durability for time-efficiency are 0.0136 and 0.0240 for version 1 and version 2 respectively. Contributions of security durability for auditability are 0.0071 and 0.0137 for version 1 and version 2 respectively. Contributions of security durability for psychological acceptability are 0.0094 and 0.0185 for version 1 and version 2 respectively. Contributions of security durability for business continuity are 0.0167 and 0.0263 for version 1 and version 2 respectively. Contributions of security durability for accessibility are 0.0043 and 0.0079 for version 1 and version 2 respectively. Contributions of

security durability for extensibility are 0.0118 and 0.0198 for version 1 and version 2 respectively. Contributions of security durability for flexibility are 0.0101 and 0.0173 for version 1 and version 2 respectively. Contributions of security durability for detectability are 0.0105 and 0.0167 for version 1 and version 2 respectively. Contributions of security durability for scalability are 0.0012 and 0.0021 for version 1 and version 2 respectively. Contributions of security durability for traceability are 0.0023 and 0.0039 for version 1 and version 2 respectively.

5.6 Conclusion

The software security area of software engineering has been largely ignored since the birth of software. There may be different reasons behind it. A long time back it was an easy task to do by applying only some passwords or installing some software. As the time passed complex antivirus software has taken place of easy to install software. The multiple connection making policy of computer makes it vulnerable for any viruses and thus making it insecure for handling personal and sensitive information. Though there has been lot of work done in field of software security to achieve maximum security in less time and cost. Security also needs maintenance. The cost and time incurred on maintenance is increasing day by day.

To reduce the maintenance time and cost and to improve security life span of software, estimation of security durability is helped in minimizing time and cost on maintenance for a specific time period. Hence, Fuzzy AHP methodology is used in this chapter to estimate the security durability of software including version 1 and version 2. Further, the assessment of security durability attributes according to their respective priorities should be followed to obtain the best possible results. Also, ratings of version 1 and version 2 are calculated using fuzzy aggregation in this chapter.

CHAPTER - VI
IMPLEMENTATION OF THE FRAMEWORK
-USING CLASSICAL MULTI CRITERIA DECISION ANALYSIS-

6.1 Introduction

A majority of organizations distinguish rapidly changing business and regulatory demands to modify how security (basically maintaining CIA) is managed during software development process [13]. Theoretical assessments are only good in papers, but in practical real time validation of theory matters a lot. Quantitative assessment is true in all engineering disciplines, including software engineering [102]. To improve the strength of security life span of a software, the proposed work presents quantitative assessment. After the implementation of the framework through Fuzzy MCDA method, this chapter is using another method, which is called Classical Multi Criteria Decision Analysis technique i.e. Classical Analytic Hierarchy Process (Classical AHP) to prove the correctness of the whole assessments and results of chapter 5. AHP is a decision aid for helping to solve unstructured problems in economics, social and information sciences. The security durability of software for both versions (version 1 and version 2) are evaluated through classical AHP to prove the accuracy of the results. Step by step method is implemented in next sections.

6.2 Evaluating the Weights of Attributes through Classical Method

In classical AHP, the process of data collection and assessment of that data is same as Fuzzy AHP but only difference is that there is no fuzzification required. Hence, the data is taken in its crisp form for classical AHP. According to the AHP process first a decision hierarchy has been developed which is same as in figure 5.2(a) from chapter 5. In the next step, pair wise matrix of expert's judgments has been developed according to the Saaty's scale of judgments which is same as shown in table 3.6.2 (a) but this method is using the numeric values directly on the behalf of TFN values [157]. With the help of the scale, linguistic values are converted into numeric values. Next step is to aggregate the pair wise comparison matrix of expert's judgments [104] while consistency ratio of the pair wise matrix is checked with the help of equations (11, 14-15) and table 3.6.2(c). Further, according to the set of attributes in the

hierarchy, the relative local weights and ranks of each set of attributes have been depicted in the table 6.2(a) to table 6.2(k).

Table 6.2(a): Local Weights of Attributes for First Level through Classical Method

	Dependability (C1)	**Trustworthiness** (C2)	**Human Trust** (C3)	**Weights**
Dependability (C1)	1	1.8180	1.9651	**0.4856**
Trustworthiness (C2)	0.5500	1	1.1087	**0.2694**
Human Trust (C3)	0.5089	0.9020	1	**0.2450**
				CR=0.00617

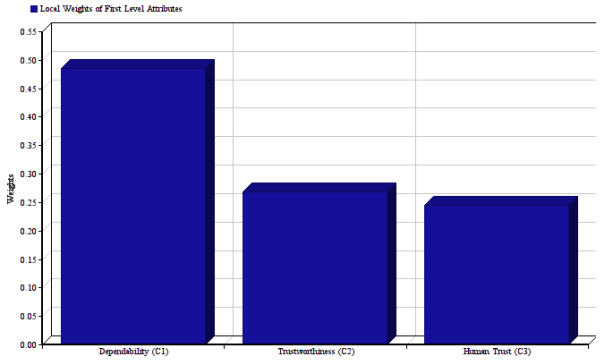

Figure 6.2(a): Graphical Representation of Local Weights for First Level through Classical Method

Table 6.2(a) and figure 6.2(a) shows the local weights of first level attributes of the hierarchy. Consistency Ratio (CR) is 0.00617 and less than 0.1. This CR value is acceptable to continue AHP analysis. This set of attributes have three attributes including dependability (0.4856), trustworthiness (0.2694), human trust (0.2450) and dependability is highest weighted factor among them.

Table 6.2(b): Local Weight of Attributes for C1 of Second Level through Classical Method

	Availability (C11)	**Reliability** (C12)	**Maintainability** (C13)	**Confidentiality** (C14)	**Authentication** (C15)	**Weights**
Availability (C11)	1	0.4395	0.9012	0.2928	0.3386	**0.0906**
Reliability (C12)	2.2753	1	3.1699	0.3657	1.0059	**0.2218**
Maintainability (C13)	1.1096	0.3155	1	0.5251	0.5220	**0.1129**
Confidentiality (C14)	3.4153	2.7345	1.9044	1	1.1486	**0.3362**
Authentication (C15)	2.9533	0.9941	1.9157	0.8706	1	**0.2385**
						CR=0.0398

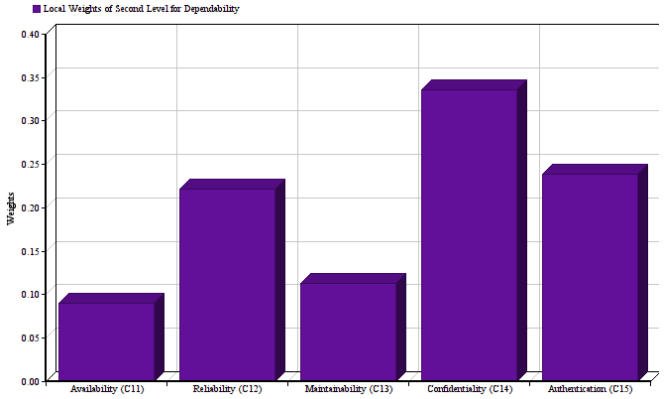

Figure 6.2(b): Graphical Representation for C1 of Second Level through Classical Method

Table 6.2(b) and figure 6.2(b) shows the local weights for C1 of Second Level attributes. Consistency Ratio (CR) is 0.0398 and less than 0.1. This CR value is acceptable to continue AHP analysis. This set of attributes has five attributes including availability (0.0906), reliability (0.2218), maintainability (0.1129), confidentiality (0.3262), authentication (0.2385) and confidentiality is highest weighted factor among them.

Table 6.2(c): Local Weight of Attributes for C2 of Second Level through Classical Method

	Availability (C21)	**Reliability** (C22)	**Maintainability** (C23)	**Accountability** (C24)	**Survivability** (C25)	**Weights**
Availability (C21)	1	0.8994	0.8831	0.7029	0.5497	**0.1499**
Reliability (C22)	1.1119	1	1.2376	0.5483	0.6710	**0.1674**
Maintainability (C23)	1.1324	0.8080	1	0.7093	0.3854	**0.1443**
Accountability (C24)	1.4227	1.8238	1.4098	1	0.5881	**0.2236**
Survivability (C25)	1.8192	1.4903	2.5947	1.7004	1	**0.3148**
						CR=0.0125

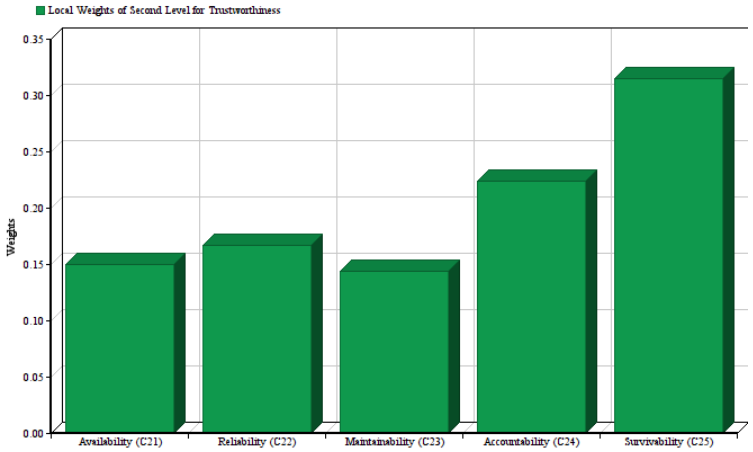

Figure 6.2(c): Graphical Representation for C2 of Second Level through Classical Method

Table 6.2(c) and figure 6.2(c) shows the local weights for C2 of Second Level attributes. Consistency Ratio (CR) is 0.0125 and less than 0.1. This CR value is acceptable to continue AHP analysis. This set of attributes have five attributes including availability (0.1499), reliability (0.1674), maintainability (0.1443), accountability (0.2236), survivability (0.3148) and survivability is highest weighted factor among them.

Table 6.2(d): Local Weight of Attributes for C3 of Second Level through Classical Method

	Reliability (C31)	**Consumer Integrity** (C32)	**Accountability** (C33)	**Confidentiality** (C34)	**Authentication** (C35)	**Weights**
Reliability (C31)	1	1.2475	1.5849	1.0118	0.9120	**0.2194**
Consumer Integrity (C32)	0.8016	1	0.9143	0.6335	0.4900	**0.1446**
Accountability (C33)	0.6310	1.0937	1	0.6575	0.6597	**0.1526**
Confidentiality (C34)	0.9883	1.5785	1.5209	1	0.6448	**0.2116**
Authentication (C35)	1.0965	2.0408	1.5158	1.5509	1	**0.2718**
						CR=0.0066

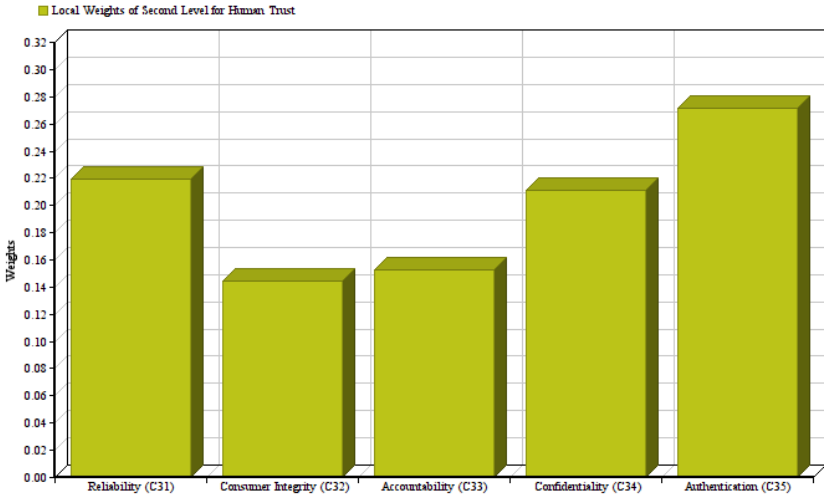

Figure 6.2(d): Graphical Representation for C3 of Second Level through Classical Method

Table 6.2(d) and figure 6.2(d) shows the local weights for C3 of Second Level attributes. Consistency Ratio (CR) is 0.0066 and less than 0.1. This CR value is acceptable to continue AHP analysis. This set of attributes have five attributes including reliability (0.2194), consumer integrity (0.1446), accountability (0.1526), confidentiality (0.2116), authentication (0.2718) and authentication is highest weighted factor among them.

Table 6.2(e): Local Weight of Attributes for C11 of Third Level through Classical Method

	Auditability (C111)	**Feasibility** (C112)	**Accessibility** (C113)	**Software Effectiveness Evaluation** (C114)	**Operational Controls** (C115)	**Weights**
Auditability (C111)	1	2.5710	1.6842	2.4385	0.5724	**0.2354**
Feasibility (C112)	0.3890	1	0.7754	0.9504	0.1953	**0.0919**
Accessibility (C113)	0.5938	1.2897	1	1.0502	0.2462	**0.1189**
Software Effectiveness Evaluation (C114)	0.4101	1.0522	0.9522	1	0.2283	**0.1021**
Operational Controls (C115)	1.7470	5.1203	4.0617	4.3802	1	**0.4517**
						CR=0.0026

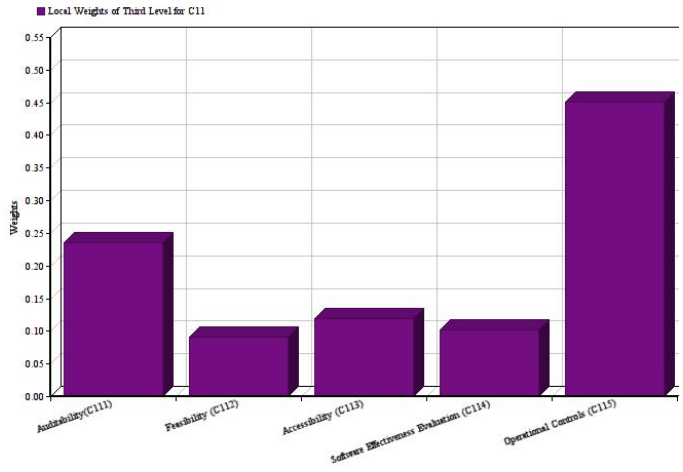

Figure 6.2 (e): Graphical Representation for C11 of Third Level through Classical Method

Table 6.2(e) and figure 6.2(e) shows the local weights for C11 of Third Level attributes. Consistency Ratio (CR) is 0.0026 and less than 0.1. This CR value is acceptable to continue AHP analysis. This set of attributes have five attributes including auditability (0.2354), feasibility (0.0919), accessibility (0.1189), software effectiveness evaluation (0.1021), operational controls (0.4517) and operational controls is highest weighted factor among them.

Table 6.2(f): Local Weight of Attributes for C12 of Third Level through Classical Method

	Feasibility (C121)	**Time-efficiency** (C122)	**User Satisfaction** (C123)	**Business Continuity** (C124)	**Weights**
Feasibility (C121)	1	2.3498	1.9575	1.5543	**0.3881**
Time-efficiency (C122)	0.4256	1	0.7860	0.7195	**0.1663**
User Satisfaction (C123)	0.5109	1.2723	1	0.8123	**0.2024**
Business Continuity (C124)	0.6434	1.3899	1.2311	1	**0.2432**
					CR=0.0006

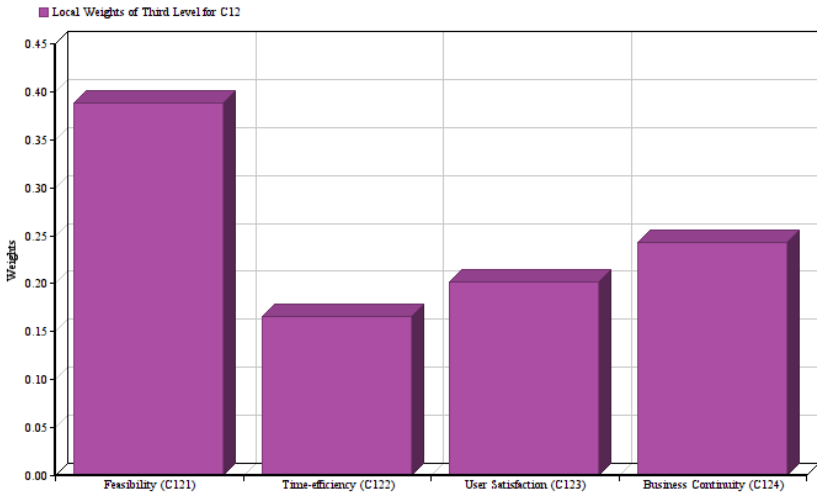

Figure 6.2(f): Graphical Representation for C12 of Third Level through Classical Method

Table 6.2(f) and figure 6.2(f) shows the local weights for C12 of Third Level attributes. Consistency Ratio (CR) is 0.0006 and less than 0.1. This CR value is acceptable to continue AHP analysis. This set of attributes have four attributes including feasibility (0.3881), time-efficiency (0.1663), user satisfaction (0.2024), business continuity (0.2432) and feasibility is highest weighted factor among them.

Table 6.2(g): Local Weight of Attributes for C13 of Third Level through Classical Method

	Auditability (131)	**Scalability** (132)	**Traceability** (133)	**Detectability** (134)	**Extensibility** (135)	**Flexibility** (136)	**Accessibility** (137)	**Time-efficiency** (138)	**Weights**
Auditability (131)	1	1.5157	0.6372	0.5743	0.2871	0.4610	1.1653	0.3238	**0.0681**
Scalability (132)	0.6598	1	0.6657	0.3936	0.3521	0.1969	0.5743	0.2076	**0.0467**
Traceability (133)	1.5693	1.5022	1	1.3195	0.4352	0.8705	1.8250	0.2091	**0.1016**
Detectability (134)	1.7413	2.5407	0.7579	1	0.9143	1.0592	1.3465	0.7489	**0.1259**
Extensibility (135)	3.4831	2.8401	2.2978	1.0937	1	0.6372	1.1095	0.3300	**0.1389**
Flexibility (136)	2.1692	5.0787	1.1488	0.9441	1.5694	1	2.5508	1.0352	**0.1769**
Accessibility (137)	0.8582	1.7413	0.5480	0.7427	0.9013	0.3920	1	0.2575	**0.0748**
Time-efficiency (138)	3.0883	4.8170	4.7824	1.3353	3.0303	0.9660	3.8835	1	**0.2671**
									CR=0.03856

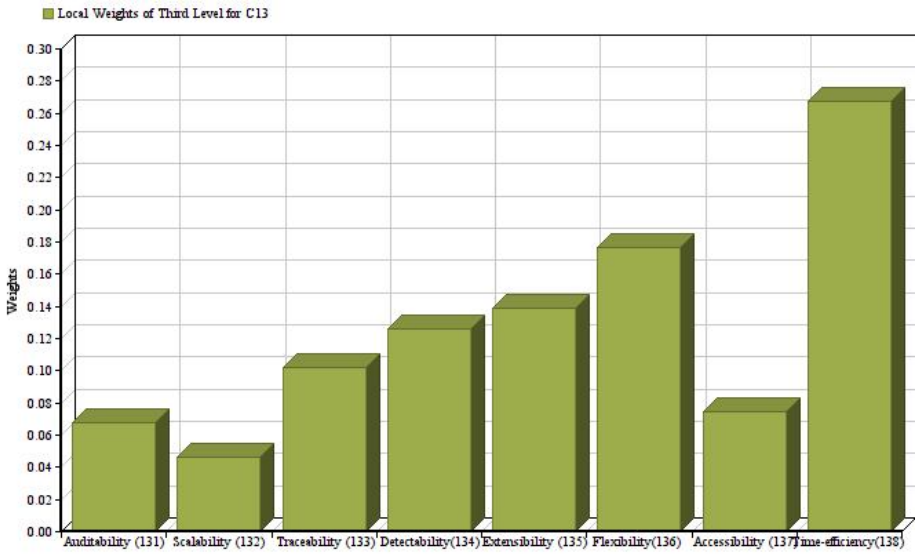

Figure 6.2(g): Graphical Representation for C13 of Third Level through Classical Method

Table 6.2(g) and figure 6.2(g) shows the local weights for C13 of Third Level attributes. Consistency Ratio (CR) is 0.0389 and less than 0.1. This CR value is acceptable to continue AHP analysis. This set of attributes have eight attributes including auditability (0.0681),

scalability (0.0467), traceability (0.1016), detectability (0.1259), extensibility (0.1389), flexibility (0.1769), accessibility (0.0748), time-efficiency (0.2671) and time-efficiency is highest weighted factor among them.

Table 6.2(h): Local Weight of Attributes for C14 of Third Level through Classical Method

	User Satisfaction (C141)	**Software Effectiveness Evaluation** (C142)	**Operational Controls** (C143)	**Weights**
User Satisfaction (C141)	1	0.8860	0.2762	**0.1793**
Software Effectiveness Evaluation (C142)	1.1287	1	0.3892	**0.2179**
Operational Controls (C143)	3.6206	2.5694	1	**0.6028**
				CR=0.0047

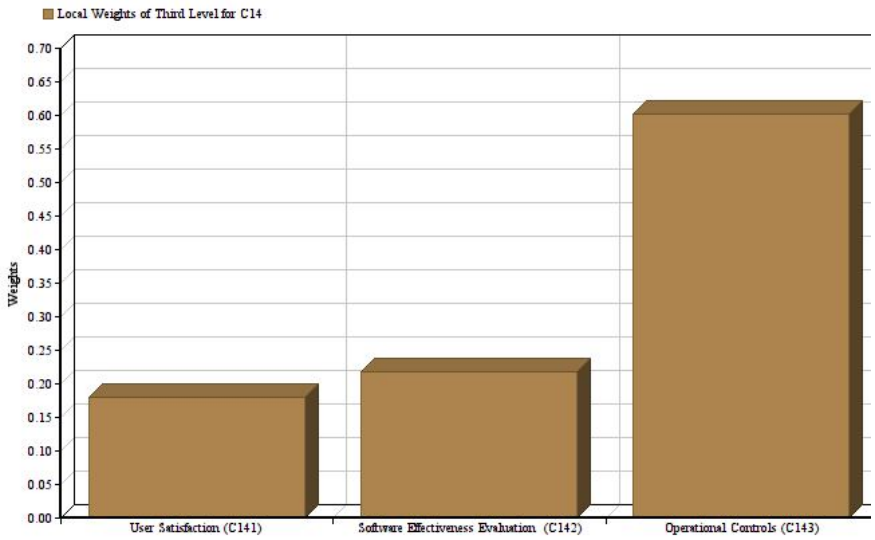

Figure 6.2(h): Graphical Representation for C14 of Third Level through Classical Method

Table 6.2(h) and figure 6.2(h) shows the local weights for C14 of Third Level attributes. Consistency Ratio (CR) is 0.0047 and less than 0.1. This CR value is acceptable to continue AHP analysis. This set of attributes have three attributes including user satisfaction (0.1793), software effectiveness evaluation (0.2179), operational controls (0.6028) and operational controls is highest weighted factor among them.

Table 6.2(i): Local Weight of Attributes for C15 of Third Level through Classical Method

	Psychological Acceptability (C151)	**User Satisfaction** (C152)	**Software Effectiveness Evaluation** (C153)	**Operational Controls** (C154)	**Weights**
Psychological Acceptability (C151)	1	1.3741	0.8360	0.3766	**0.1802**
User Satisfaction (C152)	0.7277	1	0.4208	0.2303	**0.1138**
Software Effectiveness Evaluation (C153)	1.1961	2.3764	1	0.7959	**0.2732**
Operational Controls (C154)	2.6553	4.3422	1.2564	1	**0.4328**
					CR=0.0147

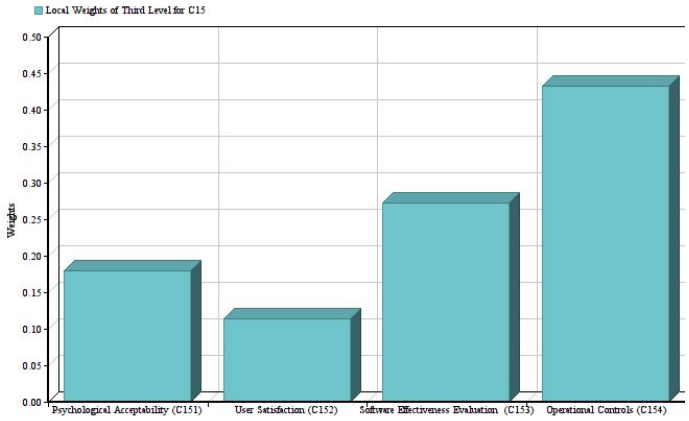

Figure 6.2(i): Graphical Representation for C15 of Third Level through Classical Method

Table 6.2(i) and figure 6.2(i) shows the local weights for C15 of Third Level attributes. Consistency Ratio (CR) is 0.0147 and less than 0.1. This CR value is acceptable to continue AHP analysis. This set of attributes have four attributes including psychological acceptability (0.1802), user satisfaction (0.1138), software effectiveness evaluation (0.2732), operational controls (0.4328) and operational controls is highest weighted factor among them. Due to repeated attributes in second level, some set of third level attributes are repeated when set of attributes considered as independently. Hence, local weights of third level attributes for C21, C22 and C23 are same as C11, C12 and C13 respectively.

Table 6.2(j): Local Weight of Attributes for C25 of Third Level through Classical Method

	Detectability (C251)	**Extensibility** (C252)	**Flexibility** (C253)	**Weights**
Detectability (C251)	1	0.9502	1.4385	**0.3632**
Extensibility (C252)	1.0524	1	1.5826	**0.3880**
Flexibility (C253)	0.6952	0.6319	1	**0.2488**
				CR=0.000019

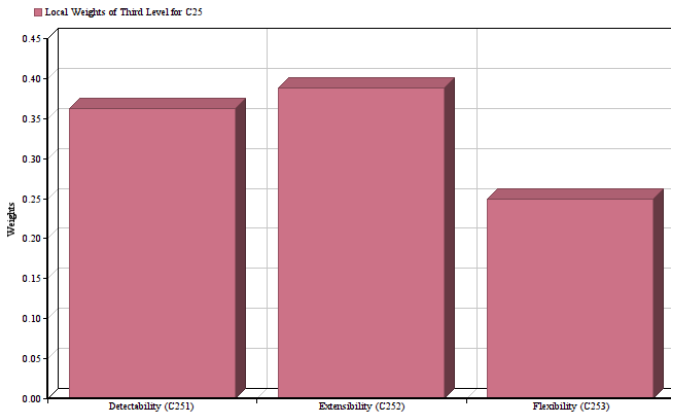

Figure 6.2(j): Graphical Representation for C25 of Third Level through Classical Method

Table 6.2(j) and figure 6.2(j) shows the local weights for C25 of Third Level attributes. Consistency Ratio (CR) is 0.000019 and less than 0.1. This CR value is acceptable to continue AHP analysis. This set of attributes have three attributes including detectability (0.3632), extensibility (0.3880), flexibility (0.2488) and extensibility is highest weighted factor among them.

Table 6.2(k): Local Weight of Attributes for C32 of Third Level through Classical Method

	Psychological Acceptability (C321)	**User Satisfaction** (C322)	**Business Continuity** (C323)	**Operational Controls** (C324)	**Weights**
Psychological Acceptability (C321)	1	1.5990	1.1118	0.7132	**0.2502**
User Satisfaction (C322)	0.625391	1	0.4480	0.3172	**0.1284**
Business Continuity (C323)	0.8994	2.2321	1	1.0564	**0.2848**
Operational Controls (C324)	1.4021	3.1526	0.9466	1	**0.3366**
					CR=0.0167

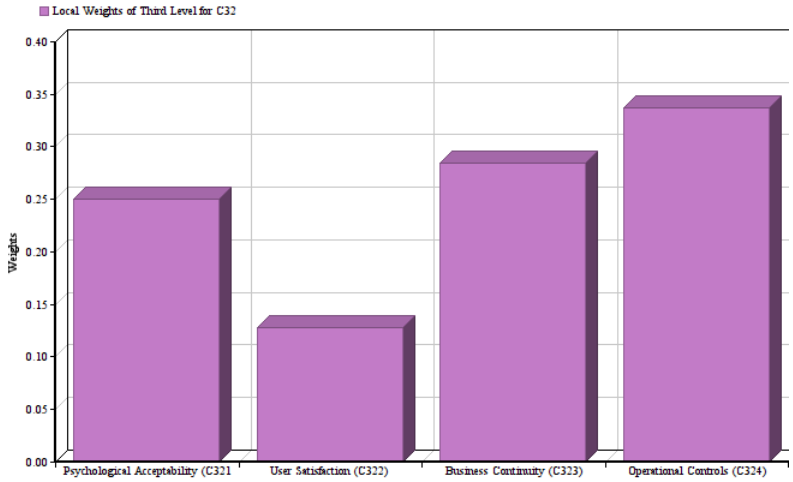

Figure 6.2(k): Graphical Representation for C32 of Third Level through Classical Method

Table 6.2(k) and figure 6.2(k) shows the local weights for C11 of Third Level attributes. Consistency Ratio (CR) is 0.0167 and less than 0.1. This CR value is acceptable to continue AHP analysis. These sets of attributes have four attributes including psychological acceptability (0.2502), user satisfaction (0.1284), business continuity (0.2848), operational controls (0.3366) and, operational controls is highest weighted factor among them. Again, local weights of third level attributes for C31, C34 and C35 are same as C12, C14 and C15 respectively. Local weights show the level wise impact of these attributes and also called

independent weights. To evaluate the weights of the security durability attributes through classical method, final weights are calculated in next portion.

Final Weight of Each Attribute through Classical Method

Final weights are also called dependent weights of security durability throughout the hierarchy. The final weights (dependent weights) of each attribute through hierarchy are shown in Table 6.2(l).

Table 6.2(l): The Final Weights of Each Criteria through Hierarchy through Classical Method

The first level	The weight of first level	The second level	Local weight of second level	The final weight of the second level	The third level	The local weight of the third level	The (Global) final weight of the third level
C1	0.4856	C11	0.0906	0.0440	C111	0.2354	0.0104
					C112	0.0919	0.0040
					C113	0.1189	0.0052
					C114	0.1021	0.0045
					C115	0.4517	0.0199
		C12	0.2218	0.1077	C121	0.3881	0.0418
					C122	0.1663	0.0179
					C123	0.2024	0.0218
					C124	0.2432	0.0262
		C13	0.1129	0.0548	C131	0.0681	0.0037
					C132	0.0467	0.0026
					C133	0.1016	0.0056
					C134	0.1259	0.0069
					C135	0.1389	0.0076
					C136	0.1769	0.0097
					C137	0.0748	0.0041
					C138	0.2671	0.0146
		C14	0.3362	0.1633	C141	0.1793	0.0293
					C142	0.2179	0.0356
					C143	0.6028	0.0984
		C15	0.2385	0.1158	C151	0.1802	0.0209
					C152	0.1138	0.0132
					C153	0.2732	0.0316
					C154	0.4328	0.0501
C2	0.2694	C21	0.1499	0.0404	C211	0.2354	0.0095
					C212	0.0919	0.0037
					C213	0.1189	0.0048
					C214	0.1021	0.0041
					C215	0.4517	0.0182
		C22	0.1674	0.0451	C221	0.3881	0.0175
					C222	0.1663	0.0075
					C223	0.2024	0.0091
					C224	0.2432	0.0110
		C23	0.1443	0.0389	C231	0.0681	0.0026
					C232	0.0467	0.0018
					C233	0.1016	0.0040
					C234	0.1259	0.0049
					C235	0.1389	0.0054
					C236	0.1769	0.0069
					C237	0.0748	0.0029
					C238	0.2671	0.0104

		C24	0.2236	0.0602	C241	-	0.0602
		C25	0.3148	0.0848	C251	0.3632	0.0308
					C252	0.3880	0.0329
					C253	0.2488	0.0211
C3	0.2450	C31	0.2194	0.0538	C311	0.3881	0.0209
					C312	0.1663	0.0089
					C313	0.2024	0.0109
					C314	0.2432	0.0131
		C32	0.1446	0.0354	C321	0.2502	0.0089
					C322	0.1284	0.0045
					C323	0.2848	0.0101
					C324	0.3366	0.0119
		C33	0.1526	0.0374	C331	-	0.0374
		C34	0.2116	0.0518	C341	0.1793	0.0093
					C342	0.2179	0.0113
					C343	0.6028	0.0312
		C35	0.2718	0.0666	C351	0.1802	0.0120
					C352	0.1138	0.0076
					C353	0.2732	0.0182
					C354	0.4328	0.0288

The table 6.2(l) summarizes the above tables and prioritizes the attributes and sub attributes according to its weightages. Through the hierarchy, the decomposition of security durability attributes has been considered in three levels viz., level 1, level 2 and level 3.

6.3 Ratings of Attributes through Classical Method

Simple aggregation method is the easiest and reliable method to calculate the average of results. Researcher took the ratings and converted the linguistic values into numerical values with the help of rating scale table 3.6.2(b) and aggregation method [109] is used to evaluate the ratings (also called objective weightages) of security durability attributes for version 1 and version 2. Further, aggregation method is used in various research areas for decision making in different fields such as decision making, rating and so on [97]. The aggregation of the ratings through classical method is discussed in next portion.

Local Ratings through Classical Method

With the help of equations (2-5), local ratings of security durability attributes is estimated. These local ratings is also called independent ratings. Further, table 6.3 (a) is showing the local ratings for version 1 and version 2.

Table 6.3(a): Local Rating of the Attributes for Level 1, 2 and 3 through Classical Method

S. No.	Characteristics of Level 1	Average Ratings	
		Version 1	Version 2
1	Dependability	0.62	0.79
2	Trustworthiness	0.64	0.84

3	Human Trust	0.60	0.82
S. No.	**Characteristics of Level 2**	**Average Ratings**	
		Version 1	**Version 2**
1	Reliability	0.72	0.81
2	Availability	0.63	0.82
3	Authentication	0.55	0.85
4	Maintainability	0.64	0.84
5	Confidentiality	0.72	0.70
6	Accountability	0.62	0.83
7	Consumer Integrity	0.64	0.90
8	Survivability	0.68	0.87
S. No.	**Characteristics of Level 3**	**Average Ratings**	
		Version 1	**Version 2**
1	Software Effectiveness Evaluation	0.60	0.75
2	User Satisfaction	0.81	0.64
3	Feasibility	0.57	0.65
4	Operational Controls	0.67	0.78
5	Time-efficiency	0.52	0.85
6	Auditability	0.60	0.58
7	Psychological Acceptability	0.58	0.72
8	Business Continuity	0.57	0.57
9	Accessibility	0.61	0.61
10	Extensibility	0.60	0.56
11	Flexibility	0.66	0.54
12	Detectability	0.56	0.59
13	Scalability	0.62	0.66
14	Traceability	0.57	0.57

Table 6.3(a) shows the local ratings of security durability attributes for level 1, level 2 and level 3 respectively. Ratings for the set of first level attributes have three attributes including dependability (0.620, 0.790), trustworthiness (0.640, 0.840) and human trust (0.600, 0.820) for version 1 and version 2 respectively and trustworthiness is highest rated factor among them. Similarly, the local ratings of level 2 and level 3 are shown in table 6.3 (a). Further, local ratings are shown the level wise impact of these attributes for version 1 and version 2. Further, local ratings also called independent ratings. To evaluate the impact of the security durability

attributes throughout the hierarchy, final ratings are calculated through classical method in next portion.

Final Rating through Classical Method

Table 6.3(b) above shows the independent ratings of every attribute at level 1, 2 and 3 by using classical method. Next step in this row is to calculate the final ratings of attributes according to their place in hierarchy. Final ratings of each attribute through classical method are shown in table 6.3(b).

Table 6.3(b): Final Ratings of Each Attribute through Classical Method

The first level	The Ratings of durability factors of the first level		The second level	Local Ratings of second level		The final Ratings of the second level		The level of the third level	The local Ratings of the third level		The final Ratings of the third level	
	Version 1	Version 2		Version 1	Version 2	Version 1	Version 2		Version 1	Version 2	Version 1	Version 2
C1	0.61	0.79	C11	0.63	0.82	0.384	0.648	C111	0.60	0.750	0.231	0.486
								C112	0.57	0.640	0.219	0.415
								C113	0.61	0.650	0.234	0.421
								C114	0.75	0.780	0.288	0.505
								C115	0.67	0.850	0.257	0.551
			C12	0.72	0.81	0.439	0.640	C121	0.57	0.640	0.250	0.410
								C122	0.52	0.580	0.228	0.371
								C123	0.81	0.720	0.356	0.461
								C124	0.57	0.570	0.250	0.365
			C13	0.63	0.84	0.384	0.664	C131	0.60	0.750	0.231	0.498
								C132	0.62	0.610	0.238	0.405
								C133	0.57	0.560	0.219	0.372
								C134	0.56	0.540	0.215	0.358
								C135	0.6	0.590	0.231	0.392
								C136	0.66	0.660	0.254	0.438
								C137	0.61	0.650	0.234	0.431
								C138	0.52	0.580	0.200	0.385
			C14	0.72	0.7	0.439	0.553	C141	0.81	0.720	0.356	0.398
								C142	0.75	0.780	0.329	0.431
								C143	0.67	0.850	0.294	0.470
			C15	0.55	0.85	0.336	0.672	C151	0.58	0.570	0.195	0.383
								C152	0.81	0.720	0.272	0.483
								C153	0.75	0.780	0.252	0.524
								C154	0.67	0.850	0.225	0.571
C2	0.64	0.84	C21	0.63	0.82	0.403	0.689	C211	0.6	0.750	0.242	0.517
								C212	0.57	0.640	0.230	0.441
								C213	0.61	0.650	0.246	0.448
								C214	0.75	0.780	0.302	0.537
								C215	0.67	0.850	0.270	0.585
			C22	0.72	0.81	0.461	0.680	C221	0.57	0.640	0.263	0.435
								C222	0.52	0.580	0.240	0.395
								C223	0.81	0.720	0.373	0.490
								C224	0.57	0.570	0.263	0.388
			C23	0.63	0.84	0.403	0.706	C231	0.6	0.750	0.242	0.529
								C232	0.62	0.610	0.250	0.430

								C233	0.57	0.560	0.230	0.395
								C234	0.56	0.540	0.226	0.381
								C235	0.6	0.590	0.242	0.416
								C236	0.66	0.660	0.266	0.466
								C237	0.61	0.650	0.246	0.459
								C238	0.52	0.580	0.210	0.409
			C24	0.61	0.83	0.480	0.655	C241	0.75	0.780	0.360	0.511
			C25	0.68	0.87	0.435	0.731	C251	0.56	0.540	0.244	0.395
								C252	0.6	0.590	0.261	0.431
								C253	0.66	0.660	0.287	0.482
C3	0.6	0.82	C31	0.72	0.81	0.432	0.664	C311	0.57	0.640	0.246	0.425
								C312	0.52	0.580	0.225	0.385
								C313	0.81	0.720	0.350	0.478
								C314	0.57	0.570	0.246	0.379
			C32	0.63	0.9	0.378	0.738	C321	0.58	0.570	0.219	0.421
								C322	0.81	0.720	0.306	0.531
								C323	0.57	0.570	0.215	0.421
								C324	0.67	0.850	0.253	0.627
			C33	0.61	0.83	0.366	0.681	C331	0.75	0.780	0.275	0.531
			C34	0.55	0.7	0.330	0.574	C341	0.81	0.720	0.267	0.413
								C342	0.75	0.780	0.248	0.448
								C343	0.67	0.850	0.221	0.488
			C35	0.61	0.85	0.366	0.697	C351	0.58	0.570	0.212	0.397
								C352	0.81	0.720	0.296	0.502
								C353	0.75	0.780	0.275	0.544
								C354	0.67	0.850	0.245	0.592

Table 6.3(b) shows that many attributes at level 2 and level 3 are same but their impact (ratings) on their corresponding higher-level attributes is different. With the help of hierarchy, dependent ratings are evaluated. With the help of final ratings and weights, security durability of software is estimated for version 1 and version 2 in next portion through classical method.

6.4 Assessment of Security Durability through Classical Method

From equation (16), security durability is assessed for two alternatives i.e. version 1 and version 2 with the help of final ratings (R_i) and weights (W_i) of attributes. The calculation of the assessment is as follows:

[0.0104 0.0040 0.0052 0.0045 0.0199 0.0418 0.0179 0.0218 0.0262 0.0037 0.0026 0.0056 0.0069 0.0076 0.0097 0.0041 0.0146 0.0293 0.0356 0.0984 0.0209 0.0132 0.0316 0.0501 0.0095 0.0037 0.0048 0.0041 0.0182 0.0175 0.0075 0.0091 0.0110 0.0026 0.0018 0.0040 0.0049 0.0054 0.0069 0.0029 0.0104 0.0602 0.0308 0.0329 0.0211 0.0209 0.0089 0.0109 0.0131 0.0089 0.0045 0.0101 0.0119 0.0374 0.0093 0.0113 0.0312 0.0120 0.0076 0.0182 0.0288] ×

$$\begin{bmatrix} 0.231 & 0.486 \\ 0.219 & 0.415 \\ 0.234 & 0.421 \\ 0.288 & 0.505 \\ 0.257 & 0.551 \\ 0.250 & 0.410 \\ 0.228 & 0.371 \\ 0.356 & 0.461 \\ 0.250 & 0.365 \\ 0.231 & 0.498 \\ 0.238 & 0.405 \\ 0.219 & 0.372 \\ 0.215 & 0.358 \\ 0.231 & 0.392 \\ 0.254 & 0.438 \\ 0.234 & 0.431 \\ 0.200 & 0.385 \\ 0.356 & 0.398 \\ 0.329 & 0.431 \\ 0.294 & 0.470 \\ 0.195 & 0.383 \\ 0.272 & 0.483 \\ 0.252 & 0.524 \\ 0.225 & 0.571 \\ 0.242 & 0.517 \\ 0.230 & 0.441 \\ 0.246 & 0.448 \\ 0.302 & 0.537 \\ 0.270 & 0.585 \\ 0.263 & 0.435 \\ 0.240 & 0.395 \\ 0.373 & 0.490 \\ 0.263 & 0.388 \\ 0.242 & 0.529 \\ 0.250 & 0.430 \\ 0.230 & 0.395 \\ 0.226 & 0.381 \\ 0.242 & 0.416 \\ 0.266 & 0.466 \\ 0.246 & 0.459 \\ 0.210 & 0.409 \\ 0.360 & 0.511 \\ 0.244 & 0.395 \\ 0.261 & 0.431 \\ 0.287 & 0.482 \\ 0.246 & 0.425 \\ 0.225 & 0.385 \\ 0.350 & 0.478 \\ 0.246 & 0.379 \\ 0.219 & 0.421 \\ 0.306 & 0.531 \\ 0.215 & 0.421 \\ 0.253 & 0.627 \\ 0.275 & 0.531 \\ 0.267 & 0.413 \\ 0.248 & 0.448 \\ 0.221 & 0.488 \\ 0.212 & 0.397 \\ 0.296 & 0.502 \\ 0.275 & 0.544 \\ 0.245 & 0.592 \end{bmatrix} = \begin{matrix} \text{Version 1} \\ \text{Version 2} \end{matrix} = \begin{bmatrix} 0.2682 \\ 0.4650 \end{bmatrix}$$

Where, values of final weights are shown in the row and values of final ratings are shown in the columns. Overall security durability is shown in table 6.4(a).

Table 6.4(a): Overall Security Durability through Classical Method

Overall Security Durability		
Security Durability	**Version 1**	**Version 2**
	0.2682	0.4650

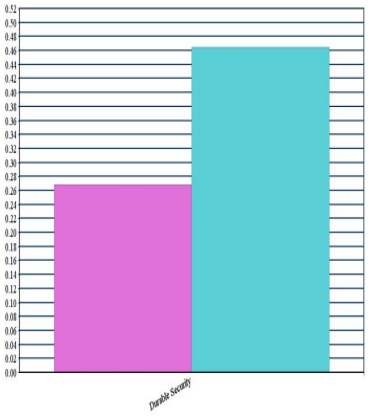

Figure 6.4(a): Graphical representation of Overall Security Durability through Classical Method

Table 6.4(a) and figure 6.4(a) are showing the values of security durability of BBAU software. Value of security durability for old version (version 1) is 0.2682 and value of security durability for modified version (version 2) is 0.4650. Again, with the help of final weights, final ratings of both version and equation (16), impact of security durability at first level is calculated which is shown in table 6.4 (b).

Table 6.4(b): Security Durability Impact at Level 1 through Classical Method

S. No.	Characteristics of Level 1	Contribution of Security Durability in Level 1	
		Version 1	**Version 2**
1	**Dependability**	0.1300	0.2204
2	**Trustworthiness**	0.0761	0.1248
3	**Human Trust**	0.0620	0.1199

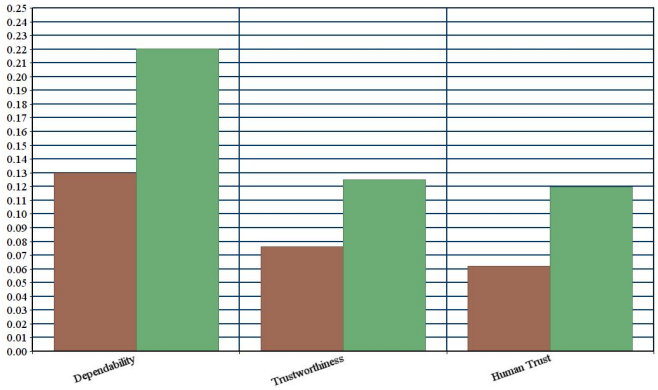

Figure 6.4(b): Graphical representation of Security Durability Impact at Level 1 through Classical Method

Table 6.4(b) and figure 6.4(b) are showing the values of security durability on first level attributes. Contributions of security durability for dependability are 0.1300 and 0.2204 for version 1 and version 2 respectively. Contributions of security durability for trustworthiness are 0.0761 and 0.1248 for version 1 and version 2 respectively. Contributions of security durability for human trust are 0.0620 and 0.1199 for version 1 and version 2 respectively. Again, with the help of final weights, final ratings of both version and equation (16), impact of security durability at second level is calculated which is shown in table 6.4 (c).

Table 6.4(c): Security Durability Impact at Level 2 through Classical Method

S. No.	Characteristics of Level 2	Contribution of Security Durability in Level 2	
		Version 1	Version 2
1	**Reliability**	0.0557	0.0852
2	**Availability**	0.0214	0.0437
3	**Authentication**	0.0438	0.0951
4	**Maintainability**	0.0214	0.0387
5	**Confidentiality**	0.0633	0.0974
6	**Accountability**	0.0320	0.0506
7	**Consumer Integrity**	0.0085	0.0178
8	**Survivability**	0.0222	0.0365

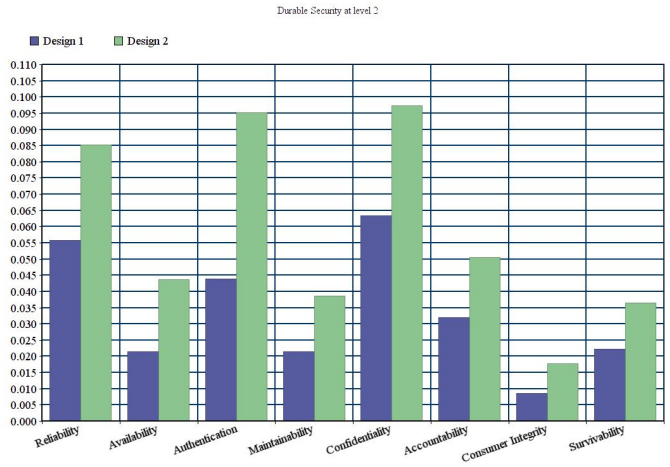

Figure 6.4(c): Graphical representation of Security Durability Impact at Level 2 through Classical Method

Table 6.4(c) and Figure 6.4(c) are showing the values of security durability on second level attributes. Contributions of security durability for reliability are 0.0557 and 0.0852 for version 1 and version 2 respectively. Similarly, contributions of security durability for second level attributes are shown for version 1 and version 2 respectively. Again, with the help of final weights, final ratings of both version and equation (16), impact of security durability at third level is calculated through classical method which is shown in table 6.4 (d).

Table 6.4(d): Contribution of Security Durability at level 3 through Classical Method

S. No.	Characteristics of Level 3	Contribution of Security Durability in Level 3 Factors	
		Version 1	Version 2
1	**Software Effectiveness Evaluation**	0.0620	0.1020
2	**User Satisfaction**	0.0351	0.0478
3	**Feasibility**	0.0219	0.0369
4	**Operational Controls**	0.0672	0.1362
5	**Time-efficiency**	0.0130	0.0229
6	**Auditability**	0.0062	0.0132
7	**Psychological Acceptability**	0.0086	0.0165
8	**Business Continuity**	0.0148	0.0230
9	**Accessibility**	0.0041	0.0074
10	**Extensibility**	0.0116	0.0194
11	**Flexibility**	0.0104	0.0176
12	**Detectability**	0.0101	0.0165
13	**Scalability**	0.0011	0.0018
14	**Traceability**	0.0021	0.0037

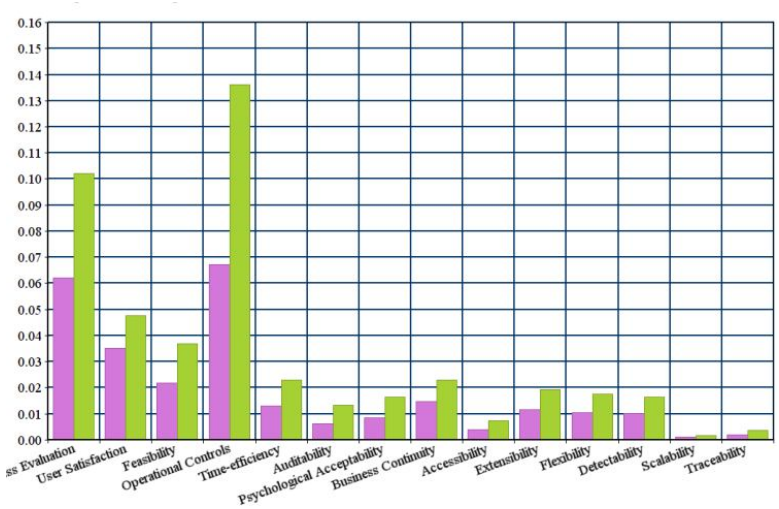

Figure 6.4 (d): Graphical representation of Security Durability Impact at Level 3 through Classical Method

Table 6.4 (d) and Figure 6.4 (d) are showing the values of security durability on second level attributes through classical method. Contributions of security durability for software effectiveness evaluation are 0.0620 and 0.1020 for version 1 and version 2 respectively. Similarly, contributions of security durability for third level attributes are shown for version 1 and version 2 respectively.

6.5 Difference between Fuzzy Method and Classical Method

Differences between the results of security durability assessment through fuzzy AHP and classical AHP methods is negligible as shown from table 6.5(a) to table 6.5 (d).

Table 6.5(a): Difference between the Results of Security Durability through Fuzzy and Classical Methods

Security Durability		
Method	**Version 1**	**Version 2**
Method 1 (Fuzzy Method)	0.2852	0.4700
Method 2 (Classical Method)	0.2682	0.4650
Difference	0.017	0.005

Table 6.5(a) shows the results of overall security durability through fuzzy AHP and classical AHP methods for version 1 and version 2. Difference between results of version 1 is 0.017 and difference between results of version 2 is 0.005 respectively. These differences between results are very low and negligible.

Table 6.5(b): Differences between Results of Level 1 through Fuzzy and Classical Methods

Contribution of Security Durability at Level 1					
S. No.	**Attributes of Level 1**	**Version 1**		**Version 2**	
		Method 1	**Method 2**	**Method 1**	**Method 2**
1	Dependability	0.1391	0.1300	0.2187	0.2204
2	Trustworthiness	0.0782	0.0761	0.1246	0.1248
3	Human Trust	0.0679	0.0620	0.1267	0.1199

Table 6.5(b) shows the results of security durability al level 1 through fuzzy and classical methods for version 1 and version 2. Difference between results of dependability for version 1 is 0.0091 and 0.0017 for version 2 respectively. Similarly, the differences between results of first level attributes are shown in table 6.5(b). These differences between results are very low and negligible.

Table 6.5(c): Differences between Results of Level 2 through Fuzzy and Classical Methods

Contribution of Security Durability at Level 2					
S. No.	**Attributes of Level 2**	**Version 1**		**Version 2**	
		Method 1	**Method 2**	**Method 1**	**Method 2**
1	**Reliability**	0.0584	0.0557	0.0903	0.0852
2	**Availability**	0.0237	0.0214	0.0433	0.0437
3	**Authentication**	0.0456	0.0438	0.0931	0.0951
4	**Maintainability**	0.0227	0.0214	0.0403	0.0387
5	**Confidentiality**	0.0696	0.0633	0.0955	0.0974

6	**Accountability**	0.0326	0.0320	0.0502	0.0506
7	**Consumer Integrity**	0.0108	0.0085	0.0214	0.0178
8	**Survivability**	0.0219	0.0222	0.0360	0.0365

Table 6.5(c) shows the results of security durability al level 2 through fuzzy AHP and classical AHP methods for version 1 and version 2. Difference between results of reliability for version 1 is 0.0027 and 0.0051 for version 2 respectively. Similarly, the differences between results of second level attributes are shown in table 6.5(c). These differences between results are very low and negligible.

Table 6.5(d): Differences between Results of Level 3 through Fuzzy and Classical Methods

Contribution of Security Durability at Level 3					
S. No.	**Attributes of Level 3**	**Version 1**		**Version 2**	
		Method 1	**Method 2**	**Method 1**	**Method 2**
1	**Software Effectiveness Evaluation**	0.0641	0.0620	0.1014	0.1020
2	**User Satisfaction**	0.0344	0.0351	0.0490	0.0478
3	**Feasibility**	0.0239	0.0219	0.0385	0.0369
4	**Operational Controls**	0.0758	0.0672	0.1310	0.1362
5	**Time-efficiency**	0.0136	0.0130	0.0240	0.0229
6	**Auditability**	0.0071	0.0062	0.0137	0.0132
7	**Psychological Acceptability**	0.0094	0.0086	0.0185	0.0165
8	**Business Continuity**	0.0167	0.0148	0.0263	0.0230
9	**Accessibility**	0.0043	0.0041	0.0079	0.0074
10	**Extensibility**	0.0118	0.0116	0.0198	0.0194
11	**Flexibility**	0.0101	0.0104	0.0173	0.0176
12	**Detectability**	0.0105	0.0101	0.0167	0.0165
13	**Scalability**	0.0012	0.0011	0.0021	0.0018
14	**Traceability**	0.0023	0.0021	0.0039	0.0037

Table 6.5(d) shows the results of security durability al level 3 through fuzzy AHP and classical AHP methods for version 1 and version 2. Difference between results of software effectiveness evaluation for version 1 is 0.0021 and 0.0006 for version 2 respectively. Similarly, the differences between results of third level attributes are shown in table 6.5(d). These differences between results are very low and negligible. The differences between all results through fuzzy AHP and classical AHP methods are very low and negligible. To evaluate the correlations between these results, next portion is discussed for statistical analysis.

6.5.1 Correlation between Results of Fuzzy Method and Classical Method

According to the Microsoft, “The correlation coefficient, like the covariance, is a measure of the extent to which two measurement variables "vary together." Unlike the covariance, the correlation coefficient is scaled so that its value is independent of the units in which the two measurement variables are expressed” [133]. Fuzzy AHP and classical AHP methods have different procedures. Further, the results are also different but very low. To statistically analyse the correlation between results, this work is taking Pearson’s correlation method [135-136] for evaluating the level wise correlations and overall correlations between results. The Pearson correlation coefficient measures the strength and direction of relationship between values of two variables. Correlation coefficient shows the impact of the relationship between two values. The scale lies between -1 and +1 [135]. The value near to -1 shows the lower bonding between values and the value near to +1 shows the tighter bonding between values. After statistical analysis, the correlation coefficients between security durability results are 0.9854 and 0.09927 for version 1 and version 2 respectively. Further, correlation coefficients for Level 1, 2 and 3 are shown in table 6.5.1 (a).

Table 6.5.1(a): Pearson’s Correlation Coefficient for Level 1, 2 and 3

	Correlations Coefficient between Results of Fuzzy and Classical Methods for Version 1	Correlations Coefficient between Results of Fuzzy and Classical Methods for Version 2
Level 1	0.9980	0.9980
Level 2	0.9973	0.9965
Level 3	0.9976	0.9993

After statistical analysis, the correlation coefficients between security durability results at level 1 are 0.9980 and 0.09980 for version 1 and version 2 respectively. The correlation coefficients between security durability results at level 2 are 0.9973 and 0.09965 for version 1 and version 2 respectively. The correlation coefficients between security durability results at level 3 are 0.9976 and 0.09993 for version 1 and version 2 respectively. The values of correlation coefficient prove the correctness of results through fuzzy and classical methods in this work.

6.6 Conclusion

Quantitative analysis of security durability is essential to measure the impact of security durability at early development process. To prove the correctness of the results (chapter 5) through fuzzy method, this chapter dealt with all the calculations through classical method and

found that the differences between results are negligible, statistically. Further, different methods are used in this work and one of them is improved method of second method.

CHAPTER - VII
EXPERIMENTAL VALIDATION

7.1 Introduction

Security of software is required for secure system because sensitive information is always at risk [4-5]. It is being very hard to find the contribution of security at early stage of software development process which has negative or positive impact on other significant aspects [12]. Further, security has always influenced the quality of software. Developers and development organizations carry loads to develop durable software with high security. Practitioners spent lots of money to deal with durable software, but unfortunately, most of the software is still non-durable and insecure [13]. Thus, practitioners are always searching new techniques or methods for evaluating and estimating the security of software services to satisfy users and giving them security assurance for a life-span [55]. In this row, assessment of security durability provides a novel vision for developing secure as well as durable software. At design time, assessment of security durability is more efficient in the relation of improvement under the aegis of Fuzzy Multi Criteria Decision Analysis methods (Fuzzy MCDA methods). The validation done in this work further helps in validating the overall performance of secure software for longer life-span.

In absence of any standard index values or details for security durability assessment, it is very hard to validate the results. Due to values of α and β, sensitivity analysis of the results is discussed in this chapter. Further, keeping in view of the importance of validation, the results of security durability assessment are validated theoretically as well as empirically. The framework and methodology is reviewed by the experts working in the area of software security. Identified experts are provided with a questionnaire defining the methodology proposed along with some questions pertaining to the objective of the methodology, its usefulness and effectiveness. For correctness of the calculation of the results, researcher used another method in chapter 6 and found that correlation between both method's results are 0.99 percent. To statistically validate the assessment as well as improvement through the suggestions, the methodology is again implemented in other

modules of the BBAU software with the help of given suggestions and ratings of attributes in other modules through the organization.

7.2 Sensitivity Analysis of the Results

The technique used to determine how independent variable values will impact a particular dependent variable under a given set of assumptions is defined as sensitivity analysis [137]. Sensitivity analysis focuses also on analyzing the effects of changes in key values of the project and depends upon one or more input variables within the specific boundaries. In chapter 5, researcher has taken the values of α and β as 0.5 and 0.5 respectively during the defuzzification. The range of these two values ranges between 0 and 1, in such a way that a lesser value indicates greater uncertainty in decision making to preferences and risk tolerance of the participants. 0.5 value for α and β is used to represent a balanced environment because the values of α and β are dependent on environmental uncertainties. This indicates that participants are neither extremely optimistic nor pessimistic about their judgments. These values will directly affect the weights of individual criteria, priority ranking and overall assessment of security durability.

If the participants involved in priority assessment have strong background knowledge on software security, the values of α and β can be readjusted to indicate confident judgments. Further, the sets of α and β values are 81 (9x9) including (0.1, 0.1), (0.1, 0.2), (0.2, 0.1), (0.1, 0.3), (0.3, 0.1) etc. The accuracy of Fuzzy AHP can be further improved by investigating the impact of α and β values toward the final results and analysis is needed in order to determine the values of α and β truthfully. That's why, to check the variations in the results, researcher has used ten combinations of α and β values for version 1 and version 2 as experiment including E1 (0.1, 0.1), E2 (0.5, 0.1), E3 (0.5, 0.3), E4 (0.5, 0.7), E5 (0.5, 0.9), E6 (0.1, 0.5), E7 (0.3, 0.5), E8 (0.7, 0.5), E9 (0.9, 0.5), E10 (0.9, 0.9) with E0 (0.5, 0.5). Further, value of α is constant for E2, E3, E4, E5 and value of β is in variation. While, value of β is constant for E6, E7, E8, E9 and value of α is in variation. The results are shown in table 7.2 (a).

Table 7.2(a): Sensitivity Analysis Due to α and β values

	Version 1	Version 2	Version 1	Version 2	Version 1	Version 2	Version 1	Version 2	Version 1	Version 2	Version 1	Version 2	Version 1	Version 2	Version 1	Version 2	Version 1	Version 2	Version 1	Version 2	Version 1	Version 2
Experiment Number	**E1**		**E2**		**E3**		**E4**		**E5**		**E0**		**E6**		**E7**		**E8**		**E9**		**E10**	
(Preferences of Participants) α	**0.1**		**0.5**		**0.5**		**0.5**		**0.5**		**0.5**		**0.1**		**0.3**		**0.7**		**0.9**		**0.9**	
(Risk Tolerance of Participants) β	**0.1**		**0.1**		**0.3**		**0.7**		**0.9**		**0.5**		**0.5**		**0.5**		**0.5**		**0.5**		**0.9**	
Security Durability	**0.4642**	**0.6906**	**0.3687**	**0.5799**	**0.3263**	**0.5190**	**0.2465**	**0.4091**	**0.2110**	**0.3555**	**0.2852**	**0.4700**	**0.2910**	**0.4579**	**0.2878**	**0.4592**	**0.2789**	**0.4605**	**0.2751**	**0.4652**	**0.2427**	**0.4185**

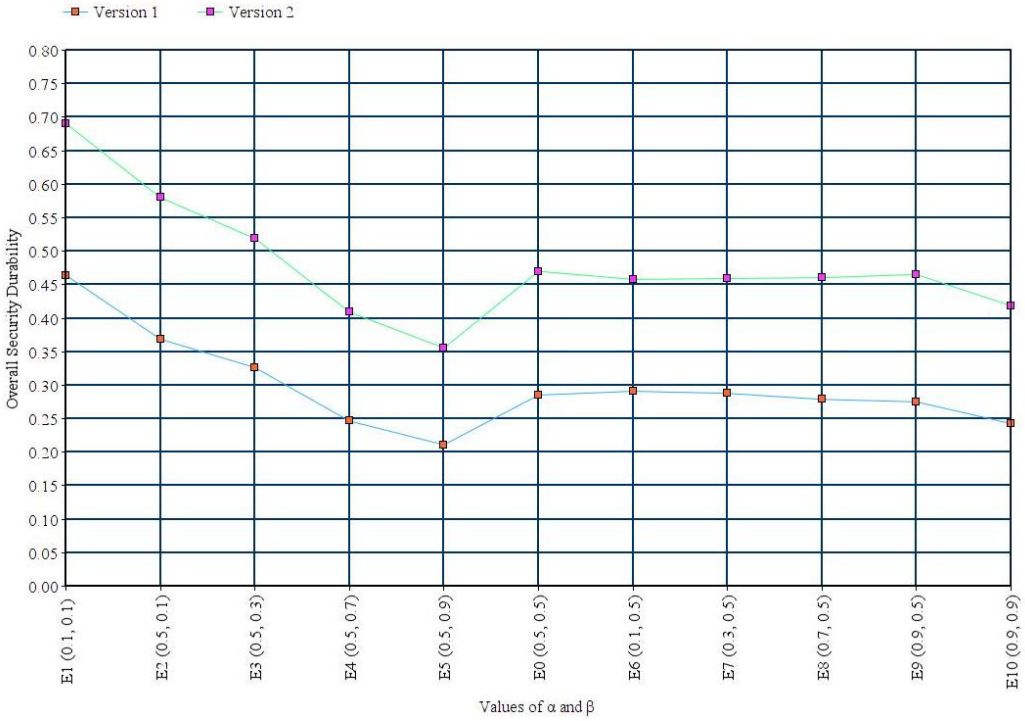

Figure 7.2 (a): Graphical Representation of Sensitivity Analysis

Table 7.2 (a) and Figure 7.2 (a) shows the variation in results due to α and β values. For, α and β values, researcher has taken the minimum values as well as maximum values for α and β including (0.1, 0.1) and (0.9, 0.9) respectively. Hence, E1 (0.1, 0.1) gives the maximum values of security durability including 0.4642, 0.6906 for version 1 and version 2 respectively but it indicates greater uncertainty in decision making. E9 (0.9, 0.9) gives the average value of security durability including 0.2427, 0.4185 for version 1 and version

2 respectively. Further, E5 (0.5, 0.9) gives the minimum value of security durability including 0.2110, 0.3555 for version 1 and version 2 respectively. All other variations due to values of α and β are shown in table 7.2 (a) and figure 7.2 (a) for version 1 and version 2 respectively. Although, E0 (0.5, 0.5) gives the concentrated values of security durability including 0.2852, 0.4700 for version 1 and version 2 respectively. The results through the values of α and β (as 0.5) indicated that a balanced environment about expert's judgments may give the best results. After going through the results of sensitivity analysis it has been determined that variation in the values of overall security durability is not negligible. Preferences of participants and risk tolerance of participants affects well on value of security durability. Further, validation of the assessment is discussed in next portion of the chapter.

7.3 Validation

Validation techniques are the best and appropriate mechanism to examine the performance and usefulness of assessment by expertise. It is also the process of finding or testing the truthfulness of something. The acceptance of an approach depends upon its validation that makes one to believe in the result of any approach. Two types of validation approaches are there including theoretical and statistical validation [138]. Theoretical validation addresses the question, if the method actually measures what it is supposed to measure whereas empirical validation addresses the question if the measure useful in the practical or experimental sense that it is related to other variables in expected ways [138]. L. Briand emphasizes on the need for thorough theoretical and empirical studies before any approach becomes widely accepted [138]. Therefore, it can be said that an approach is valid if it successfully undergoes both kinds of validation i.e. It should be carried out that the measure actually measures what it claims to measure and there are various convincing evidence that it is useful. Emphasizing the need of validation, Malvin V. Zelkowitz of NIST/ ITL says 'without a confirming experiment, why should industry select a new method or tool' [139].

Statistics provides a way to collect and organize the data. Statistical tools are used to process and analyze the data in an experiment. Statistical interpretation of the result of an experiment is one of the ways to accept or reject the outcomes of any research. It involves proposal of null hypothesis followed by data collection, data arrangement, experiment and

inference drawn from the result of the experiment. On the basis of some set criteria provided by the statisticians, the null hypothesis is accepted or rejected. Rejection of the null hypothesis leads to the acceptance of the research outcome or vice-versa.

7.3.1 Theoretical Validation

Theoretical validation is the most basic form of validation. It serves as the prerequisite to demonstrate the usefulness of a measure or empirical validation [265]. Theoretical validation requires that an analyst has in depth understanding of the concept being measured. Researcher has identified 30 practitioners from India and abroad working in the area and communicated with them for the review of the proposed work. Out of 30 experts, 20 experts responded with their valuable comments. Most of the experts are in the view that a proper structure of security durability can enhance the strength of CIA and optimize maintenance time and cost for a service life span. Almost all of the experts strongly agreed with the fact that security of software services are well affected by three attributes including trustworthiness, human trust and dependability. Experts have a strong opinion that maintenance time and cost are increasing due to the lack of durable security services. Also, lot of experts agreed with that security durability are well affected by other direct or indirect attributes including confidentiality, availability, authentication and others attributes with positive or negative impact. Therefore, proposed security durability assessment framework may be successfully used to evaluate and improve the life span of security of software services. Some of the critical observations/ suggestions made by experts are as follows:

- Without technical implementation, the concept of a framework and this assessment does not make more sense.
- Validation of the assessment will reflect its actual use.
- Implement the framework with assessment on industrial software projects.

All the suggestions are incorporated by the researcher. The framework is implemented and it is validated too.

7.3.2 Statistical Validation

Statistical validation is the process by which it can be established that a measure is useful in the sense that it is related to other variables as expected in theory. For the purpose of

statistical validation, two experiments are carried out; pre tryout and tryout. Pre tryout involves a small set of data. If analysis of the results of pre-tryout are satisfactory, try out is carried out on a larger set of data. Satisfactory results of try out conclude the acceptability of the assessment.

Design Module of an Experiment

Experiments are used for testing and exploring a given theory. Through experiments theoretical predictions are tested against reality [138]. Experimentation and data collection are the tools by which theories are validated. The goal of an experiment is to collect sufficient data in order to obtain a statistically sufficient result [138-139]. For the purpose of validating the proposed framework for security durability assessment and improvement, the experiments are performed. In pre-tryout, one module of BBAU software design is taken as input.

With the help of priorities of security durability attributes, researcher assessed the security durability of a module i.e. version 1 and then according to the suggestions, the given module is modified by the developers i.e. version 2. A comparison of assessment values shows that the security durability of version 2 after modifying is more durable than the old one. After analyzing the result of pre-tryout, since no significant changes are noticed, a tryout is carried out with the larger set of data. Ten modules of BBAU software are taken as input. The same procedure is repeated for ten modules. On the basis of the outcome, statistical interpretations are made.

Pre-Tryout

The pre-tryout has been carried out on a module of BBAU software. Security durability of version 1 and version 2 are calculated in chapter 5. The difference of security durability between version 1 and version 2 are shown in table 7.3.2 (a).

Table 7.3.2 (a): Improvement in Security Durability

	Version 1 (Old Version)	Version 2 (Modified Version)	Security Durability Improvement (In Percentage)
Security Durability	0.2852	0.4700	39.32 %

Hence, it can be said that suggestions and framework imposed on the version 2 functioned well and security durability is improved as 39. 32 %, as table 7.3.2 (a) shows.

Review & Revision

Results obtained from pre-tryout are analyzed. A critical review of the outcome strengthens the acceptability of the proposed framework and usability of the assessment. For the revision of the calculation, classical method is used in chapter 6 and got the 99 % correlation between fuzzy and classical method. Therefore, the researcher adapted the same framework for further tryout with a large set of data.

Tryout

Statistical validation is an ongoing activity and hence there are degrees of validity: the more evidences are there, the more valid is an approach [138]. For collecting more and more evidences for validation of the proposed framework, a tryout is carried out following the pre-tryout. The tryout contains ten modules of the same design. These are the modules developed by organization's team 1 and rated by team 2, again. After assessment through the researcher, the impacts of security durability of ten modules are shown in table 7.3.2 (b).

Table7.3.2 (b): Reassessing the Security Durability for Ten Modules

Software Module	Version 1 (Old Version)	Version 2 (Modified Version)	Security Durability Improvement (In Percentage)
Module 1	0.2748	0.3176	15.57 %
Module 2	0.286	0.3599	25.83 %
Module 3	0.289	0.3456	19.58 %
Module 4	0.3026	0.3267	7.96 %
Module 5	0.3178	0.3612	13.65 %
Module 6	0.3456	0.4223	22.19 %
Module 7	0.3278	0.3623	10.52 %
Module 8	0.2357	0.3267	38.60 %
Module 9	0.3425	0.4123	20.37 %
Module 10	0.3879	0.4523	16.60 %

Statistical Analysis

Statistics is a mathematical tool used for gathering, organizing, analyzing and interpreting numerical data. For the purpose of showing statistical significance or validation of the proposed framework, statistical analysis is carried out on ten modules. As the sample size is small, the two tailed t-test is applied for finding out the level of significance and rejection of the null hypothesis. Since the rejection or acceptance of a null hypothesis is based upon either (0.05) alpha (α) or (0.01) alpha (α) level of significance for one tailed or two tailed test, (0.05) alpha (α) level of significance for a one tailed test is taken for rejection of the null hypothesis. The complete process of the following statistical analysis is summarized as: the first step starts with the formulation of null hypothesis and alternate hypothesis. The values of version 1 and the values of version 2 are put under statistical analysis to draw the conclusion that whether there is a significant difference between the pretreatment data and the post treatment data. The obtained t value will determine whether to reject the null hypothesis and accept the alternative hypothesis [140].

Hypothesis Testing

A null hypothesis reflects that there is no significant relationship between two or more parameters [140] whereas alternate hypothesis affirms the relationship. Rejection of a null hypothesis provides a stronger base to accept the relationship or to accept the alternate hypothesis. Following null and alternate hypothesis were made for the purpose of validation of the proposed framework:

Null Hypothesis (H_{01}): Security durability based suggestions using security durability assessment cannot help to assess and improve the life span of security.

***Alternative Hypothesis (H_{11})*:** Security durability based suggestions using security durability assessment can help to assess and improve the life span of security.

Statistical Interpretation

By observing values of security durability in table 7.3.2 (b), it can be inferred very easily that the suggestions for all the modules have worked well. The values of security durability for different modules of version 2 are relatively more than the value of security durability for different modules of version 1. By observation, it seems that the treatment worked well. The values showed that the security durability in all the ten modules was assessed and

hence the security durability was improved. The initial claim that security durability assessment framework is able to assess and improve proved true. A graphical representation of comparative study and improvement is showing in figure 7.3.2 (a) for version 1 and version 2 respectively. But it is not over; this alone will not be able to prove acceptability of the assessment. To make the assessment derived from it, acceptable or for validation of the approach, it must be verified whether the difference in the values in version 2 is due to the given suggestions or it is just a sampling error. All in all, the level of significance of the proposed framework must be computed. While studying inferential data analysis, it was found that the t- test for the situation given below is appropriate for the purpose: 'When the same group of individuals takes a pretest then the group is exposed to a treatment. The group is again tested after treatment to determine whether the influence of the treatment has been statistically significant as determined by mean gain scores [225-265].' The t-test was carried out for drawing level of significance of the approach.

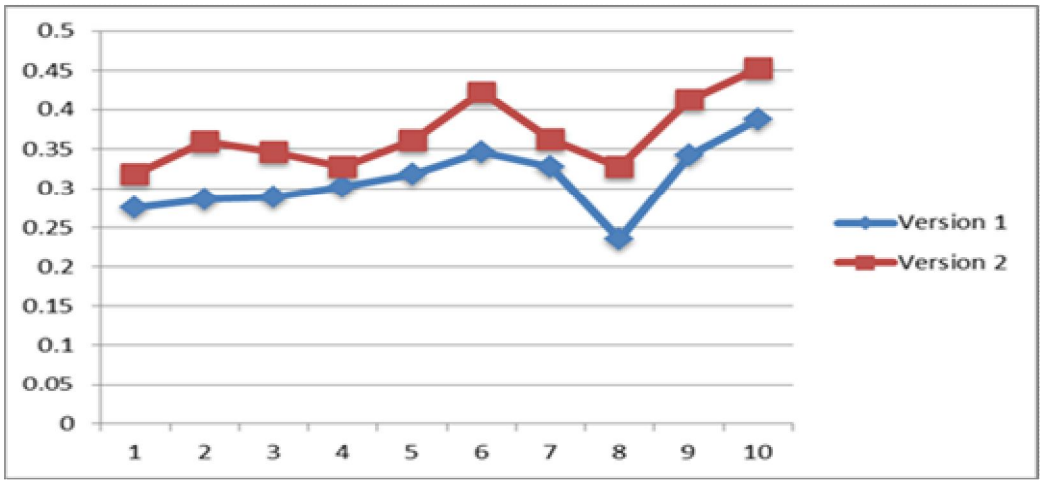

Figure 7.3.2 (a): Graphical Representation of Values of Security Durability for Different Modules

Level of Significance

To find out the significance of the difference between the means of values of version 1 and values of version 2, the means of both version 1 and version 2 values are calculated as shown in table 7.3.2 (c). Pearson coefficient of correlation comes out to be 0.887. The coefficient shows that the values of version 1 before researcher's suggestions and values of version 2 after researcher's suggestions are highly correlated. The degree of freedom is 9 for values of version 1 and version 2. For application of the t-test in the scenario, homogeneity of variances i.e. F value must be tested. The homogeneity can be obtained by dividing the larger variance by the smaller. The large variance is 0.001842 for version 1 and the smaller one is 0.00206 for version 2. The larger to smaller ratio yields F value as 0.89. Since F value is less than 1.83 (the F critical value for 2 variances of degree of freedom 9), it is concluded that the variances are

homogeneous. This test provides the ground for applicability of t-test. The t value comes out to be 2.26. As the value exceeds the t critical value of 1.83 for a one tailed test at the 0.05 level for 9 degree of freedom, the null hypothesis H_{01} is strongly rejected and the alternate hypothesis H_{11} is accepted. Hence, it is validated that security durability assessment framework can be assessed and with the help of the suggestions, security durability can be improved.

Table 7.3.2 (c): t-Test for Security Durability Improvement Data Analysis

t-Test for Security Durability							
	Mean	**Std. Deviation**	**Std. Error**	**No. of Samples**	**Pearson Coef.**	**Degree of Freedom**	**t- Values**
Security Durability (Version 1)	0.311	0.043	0.002	10	0.887	9	2.26
Security Durability (Version 2)	0.369	0.045	0.003				

7.4 Conclusion

Acceptance of any new approach by society or industry depends upon validation of that approach. It is the validation which proves the usefulness of the approach in society or in industry. For testing the usefulness of the framework for security durability assessment, a systematic validation is carried out. Initially, for the purpose of the theoretical validation, expert review was conducted. The framework was reviewed by various experts in the area and was found to be satisfactory. As a second step, statistical validation is carried out. Statistical validation involves pre-tryout and tryout. Pre-tryout involves a small set of data whereas tryout involves a larger set. The pre-tryout is carried out on a module of BBAU software design.

After, a successful pre-tryout leads the researcher to the next step i.e. tryout. The tryout is carried out on ten modules. The modules are analyzed and the values for version 1 and version 2 are computed. The values for version 1 and version 2 have undergone statistical analysis to establish the fact that framework has successfully assessed the security durability and have improved. The t-test is carried out and it is found that the t-values obtained by computation performed on values of version 1 and version 2 are exceeding the t-critical values. Hence, the null hypothesis formulated at the beginning of statistical

analysis, are rejected one by one and alternative hypothesis are accepted. Researcher claim that security durability assessment framework is able to assess the security durability and suggestions are able to improve the design for improving security life span.

CHAPTER - VIII
SUMMARY AND CONCLUSIONS

8.1 Introduction

Development of software during twenty-first century has created new challenges among all developers and users. Security has become a crucial challenge for any software product [4]. It is found that 73% of the total time and cost of the software development are being consumed on security maintenance [12-14]. The situation is expected to become worst in future. To overcome the issue, development organizations should focus on the longer secure life of software rather than maintaining it [52-54]. Hence, there is need to develop durable as well as secure software for its longer use. Without paying attention on durability at the time of software development, the security may start failing after deployment (immediately or after a time period). Ignoring durability may badly affect service life of secure software. Secure software with poor durability is likely to fail in highly competitive market [11-13]. To develop security durability cost-effectively, there is a need to investigate the relation between durability and its characteristics with security during early stage of software development.

To achieve the goal first a comprehensive review on the available literature signifying the need of security durability is done. The review appeals need of security durability to be embedded in security while development of software. A framework is proposed to accomplish the needful task which is achieving security durability. The factors affecting security durability and security of software were identified through literature review, and a survey on software security estimation and secure design factors. The development of SD^f was preceded by a thorough literature review on the identification of attributes that affect the factors of first level and security durability as well. Relationship between the security durability, its first level factors and second level factors are established. A mapping or hierarchy is created by these relationships of factors affecting each other which is helpful to estimate the impact of security durability in future. A multiple criteria decision-making method, i.e. Fuzzy AHP is used to estimate the impact of security durability on version 1 and version 2. Priority wise ranking of security durability attributes have been identified with the help of weights. With the help of the prioritization of security durability attributes, guidelines/suggestions for improvement of security durability is given. To evaluate the impact of the suggestions, this work has taken two

version of BBAU software including version 1 and version 2. Further, version 2 is modified by developers with the help of given suggestions. With the help of rate of version 1 and version 2, security durability is assessed. Theoretical, as well as empirical validation is done at the end to check the result for its significance.

8.2 Significant Contributions

The current study produces major contribution in the area of software security and security durability attributes identification including many macro levels direct or indirect findings. The estimation practice at early stage is beneficial for secure and durable software development. The assessment of security durability provides guidelines to develop secure and durable software. An assessment of security durability and suggestions produced with the help of importance of security durability attributes, revealed many things including the need of security durability in the modern era according to changing needs of user. Researcher made an attempt for the assessment of security durability having sound bearing in the literature and context. Therefore, researcher made an effort for the same to come up with the suggestions and framework SD^f. The framework comprises of five phases namely Factor Identification Phase, Classification Phase, Assessment Phase, Validation Phase and Packaging Phase with an additional common step of review and revision. On the whole SD^f is highly perspective in nature and may assure the development of good security durability assessment model.

The next phase of work contributed on different security durability estimation methodology in order to further illustrate the proposal executing every step depicted in the framework. To help the security engineers, ultimate objective is of having a quantitative assessment of security durability through MCDM techniques. The proposed model clearly follows the mentioned order of execution to achieve the target. Quantitative evaluation is helpful in deciding the high prioritized attributes to be considered for achieving high durability in security.

In order to provide the significant and improved measurement of security, it is required to correlate security durability attributes with its other common security attributes such as confidentiality, availability, authentication etc. It is evident from literature survey that there is no known complete and comprehensive work to assess security durability and its attributes at design phase. The proposed model, for the quantitative assessment of security durability, has been validated through statistical analysis. It is apparent that this methodology can be used effectively in assessing the life span of security and minimizing the cost and time spent over

maintenance of security and flaws occurring time to time. Statistical analysis has been made to strengthen the claim that expert's views are considerable while estimating the security durability in the proposed model. Empirical validation is carried out using other methodology to assess the security durability of version 1 and version 2. In this work Fuzzy AHP has been presented as major contribution towards assessing the security durability of software. Also the developed framework (SD^f) and suggestions are the other contribution towards this work.

8.3 Research Findings

Some of the preliminary research questions raised in chapter I for the study were identified and addressed directly or indirectly, during the course of the study. These questions are repeated and the finding in relation to each are answered according to the posed questions.

- **Are there any problems in the way of organizations to perceiving software security?**

The research has studied on the literature and it has been found out that organizations are not capable of achieving the security for longer time with regular maintenance period. For maintaining the software security, the organizations and its developers are always working on the newer versions of security design.

- **What are those problems?**

During the literature review of relevant work and best practices, researcher found the problem of maintaining the security for longer time is the global reason for diminishing software durability. Due to increasing maintenance cost and time users move to another option of software. When the software goes under maintenance, then the whole business of organization stops for some time and security becomes a main reason of maintenance. This harms their market value as well as business ethics. Sensitive data-based software such as defense sector, banking sector and educational software needs security for specific life span. Hence, the problem of optimal maintenance to such kind of software needs to be addressed.

- **How to minimize or optimize the security maintenance cost and security maintenance time for improving life span of security as well as software services?**

Developers are working for years to improve the maintenance process of software security. When they find no way to improve duration of security, they just move to the other processes such as redesigning, which further complicates the development process. To optimize the

security maintenance cost and reduce the time consumed, there is need to design security for longer duration.

- **What are the factors that directly influence security of software?**

The study has demonstrated that security is multidimensional, composed of several attributes that, in combination tell us not only about resistance to attack but also about the speed of return to a stable, functioning state. No matter how we decompose security characteristics, it is not clear how to combine or compose the measurable into a single number or representation of overall system security. During this research work, numerous security attributes have been identified that affect security directly or indirectly and one of them is recognized as security durability. On the basis of regress reviews of literature regarding, researcher has identified an attribute, which could be used to enhance security durability in software.

- **What are the factors affecting durability of software?**

As per the nature of security durability, the three main factors of durability have been identified including dependability, human trust and trustworthiness. Further, these factors also affect security as well. The relationship between these three major factors of security durability has been identified and depicted as a hierarchy.

- **Is there are any relation between security and durability?**

Yes, there is a direct relationship between security and durability. This thesis signifies the relationship between security and durability through a map shown in chapter 3.

- **How can we relate to security with durability?**

Security of software can be enhanced by focusing the other factor including durability. As described in chapter 1 of this work, software security affects the duration of the service life of software. Durability, in terms of software is the time period during which software gives services to consumers. Further, durability also affects the life span of security services. This statement strengthens the fact that there must be an attribute which relates to security directly i.e. durability. In this concern, durability has been considered as one of the supporting attributes for maintaining and improving the CIA. Hence, it seems that security accent has changed to the durability of the software. Security is directly or indirectly involved in the service life of the

software. Durability is further directly or indirectly involved in the security of software and vice-versa.

- **Is there any standard mechanism available for assessment of security durability?**

Security is multidimensional, emergent and irreducible concept and the unfortunate side effect of security produces inherent more complexity in design. No, there is no standard mechanism available for assessing the security durability. Quantitative evaluation of security durability is a vital process. Therefore, it is viable to develop a perspective framework that is useful for security durability assessment. Through the literature review, researcher has not identified any framework or mechanism that quantifies security durability. Researcher made a contribution in this regard to develop and validate a perspective framework that quantifies security durability of software.

- **Is it possible to estimate security durability at early stage of software development?**

Estimating security durability at early stage of development helps in achieving higher security durability at the end of development. Estimation during development phase helps in making changes in upcoming design as per the estimation policy and guidelines. Further early estimation of security durability is possible which is well defined in the thesis work.

- **Can we get a mechanism, which may be used in early stage of the system development life cycle to estimate the security durability successfully?**

Some existing mechanism can help depict our system security and resistance. This concept agrees that early estimation of security support to establish healthy conceptual building blocks with a reduced amount of effort. The adaptation of security durability concept is expected to help produce minimum and cheaper maintenance process for a life span. The main structural mechanism of this paradigm is namely dependability, human trust and trustworthiness are the keys to foster security durability. It is feasible to identify the set of security durability factors.

- **Can we develop a security quantification model targeting durability?**

Early phases of development may be used to improve security durability with its characteristics. Therefore, assessment model may be used at these levels to estimate security durability of software. Hence, a model with the set of guidelines may be developed and used to estimate the

security durability of the software. Assessment model has been developed on the basis of established relationship between durability factors and security attributes and validated through proper data set for model acceptance. The detail discussion has been covered in chapter VI and VII.

- **How can we improve the software life span through security durability estimation?**

As per the literature studied in the Chapter II, it reveals some facts about software durability and also that software durability enhances the life span of software. This infers that estimation of security durability may help in decaying the cost and time incurred on maintenance of security. Hence security durability estimation might improve the software life span by empowering its security life.

- **How can we improve the estimated security using developed models and guidelines?**

The security can be improved using concept of security durability. The impact study of durability consideration while designing security provides the required guidelines for security durability assessment and improvements. The developed security durability models are helpful to implement the security improvement guidelines for security evaluation and improvement.

- **What should organizations do in order to develop secure as well as durable software?**

The developed guidelines, in this work has been applied to BBAU software version 2 to mark the improvements in security durability of software. The organizations should follow these model and guidelines while developing their software to enhance the security durability of software. Also, organizations might give their suggestions and feedback to improve the framework and guidelines developed by researcher.

8.4 Other Findings

During the research work, researchers are also focusing another way of examination for maintaining CIA to improve security durability. Some security risks affect the durability of security during use of software services. Based on security risks, other findings through the researcher are as follows:

- Critically Reviewed the Security Risks during Software Development Process [141]

- Based on the Review of Literature Survey and Best Practices, Prepared a Checklist for Security Risk Management Process [141]
- Developed a Framework for Security Risk Management at Early Stage of Software Development Life Cycle [142]
- Identified and Classified the Security Risks that Affects the Security Design during Software Development Process [143]
- Prioritized these Security Risks through Analytical Hierarchy Process [144]
- Validated the Priority of Security Risks through Theoretical and Statistical Analysis [144]
- Facilitated the Priority based Security Risks during Software Development Process [144]
- Proposed a Model of Adaptive Neuro Fuzzy Inference System for Estimation and Prediction of Security Risks [145]
- Estimated the Security Risks through Adaptive Neuro Fuzzy Inference System with the help of MATLAB and MINITAB Tools [145]
- Compared the Proposed Model with Stepwise Regression Model [145]
- Validated the Proposed Model through Theoretical and Statistical Analysis [145]
- Projected the Security Risks through Proposed Model using MATLAB and MINITAB [145]

8.5 Impact of the Study

On the successful completion of the study, the researcher found that early security durability estimation is highly desirable in the area of secure software development. The knowledge gained from the above study may directly or indirectly contribute to prove the significance in the following manner:

- The developed framework may be used to validate other available theories which do not get the appropriate place in the literature due to lack of their theoretical and empirical validation.
- The developed framework and guidelines provide step by step procedure to quantify security durability attributes at early stage of development life cycle.
- The developed models are validated using Version 1 and Version 2 of BBAU entrance software. The model's ability to estimate overall security from design information, at least

for the different projects being used to estimate security durability and statistical analysis reported that model has been found significantly correlated.

- The proposed model may be used effectively in monitoring security durability at design phase.
- The durability attributes are helpful to affect the overall security durability ranking of the software or application.
- The development guideline proposed here helps developers of the organizations to form security of software embedded with durability.

8.6 Future Work

The model proposed to assess the security durability of software using three primary key attributes including dependability, human trust and trustworthiness. These attributes are highly significant and correlated with other attributes of security durability. The security durability assessment model has been validated, but its utility may be analyzed for larger set of data. Variation due to alpha and beta is also discussed. The assessment of security durability may provide help to developers to design the security, durable and help to maintain CIA for a life span. Test cases may be produced in the form of developer's manual for testing early security durability based on the results of the model. Different implementations of the proposed framework (SD^f) are possible. Some suggestive measures may be made to the development team to revisit the design to achieve the set of security indices related to security durability.

8.7 Conclusion

The latest issues of the computers and software related research is achieving long life span of security during software development process. The aim of the study is to assess the security durability at early stages of development. For the purpose, the framework (SD^f) integrates security attributes and durability attributes, prioritizes the attributes based on their impact on security durability and produces suggestions for developers. Through applying the suggestions on old version of BBAU software, author assessed the security durability of both old version (Version 1) and modified version (Version 2). The framework produced here may help to evaluate the security durability of software. With the help of this research, researcher may help to facilitate new activities and ideas for secure and durable software development.

References

1) Tekinerdogan B., Sozer H., Aksit M., (2008), Software Architecture Reliability Analysis using Failure Scenarios, Journal of Systems and Software, Volume 81, Issue 4, pp. 558-575.
2) Subashini S., Kavitha V., (2011), A Survey on Security Issues in Service Delivery Models of Cloud Computing, Journal of Network and Computer Applications, Volume 34, Issue 1, pp. 1-11.
3) Boegh J., (2008), A New Standard for Quality Requirements, IEEE Software, Volume 2, pp. 57-63.
4) McGraw G., (2006), Software Security: Building Security In, Volume 1, Addison-Wesley Professional.
5) SaaS Industry Market Report: Key Global Trends & Growth Forecasts (2018), Available at: https://financesonline.com/2018-saas-industry-market-report-key-global-trends-growth-forecasts/ Last Visit on 04 Sep 2018.
6) New Data: Software as a Service Industry Revenue up 23% This Year as Shift to the Cloud Continues (2017), Available at: https://www.geekwire.com/2017/new-data-software-service-industry-revenue-23-year-shift-cloud-continues/ Last Visit on 05 Sep 2018.
7) CA Veracode Report (2018), Available at: https://techbeacon.com/sorry-state-software-security-secure-development-key, Last Visit Oct 22 2018.
8) Dehaghani S. M. H., Hajrahimi N., (2013), Which Factors Affect Software Projects Maintenance Cost More?, Acta Informatica Medica, Volume 21, Issue 1, pp. 63.
9) Dalton M., Kannan H., Kozyrakis C., (2007), Raksha: A Flexible Information Flow Architecture for Software Security, ACM SIGARCH Computer Architecture News, Volume 35, Issue 2, pp. 482-493.
10) Carr N. G., (2003), IT Doesn't Matter, Educause Review, Volume 38, pp. 24-38.
11) Kelty C., Erickson S., (2015), The Durability of Software, Meson Press, Germany, Volume 1, Issue 5, pp. 1-13.
12) **Kumar R., Khan S. A., Khan R. A., (2015), Revisiting Software Security: Durability Perspective, International Journal of Hybrid Information Technology (SERSC), Volume 8, Issue 2, pp.311-322.**
13) Ensmenger N., (2014), When Good Software Goes Bad: The Surprising Durability of an Ephemeral Technology. In MICE (Mistakes, Ignorance, Contingency, and Error) Conference. Munich, pp.1-16.
14) Crossler, R., Bélanger F., (2014), An Extended Perspective on Individual Security Behaviors: Protection Motivation Theory and a Unified Security Practices (USP) Instrument, Advances in Information Systems- ACM SIGMIS, Volume 45 Issue 4, pp. 51-71.
15) Praus F., Kastner W., Palensky P., (2016), Software Security Requirements in Building Automation, Sicherheit 2016 - Sicherheit, Schutz und Zuverlässigkeit, pp. 217-228.

16) Purdy A., (2016), The Global Cyber Security Challenge, Technical Report of Huawei Technologies, 2016.
17) Gray, D., Allen, J., Cois, C., Connell, A., Ebel, E., Gulley, W.,Wisniewski, B. D. (2015), Improving Federal Cyber Security Governance through Data Driven Decision Making and Execution, Technical Report - CMU/SEI-2015-TR-011, Software Engineering Institute, Carnegie Mellon University United States.
18) Kumar R., Khan S. A., Khan R. A., (2014), Software Security Durability, International Journal of Computer Science and Technology, Volume 5, Issue 2, pp. 23-26.
19) **Kumar R., Khan S. A., Khan R. A., (2017), Secure Serviceability of Software: Durability Perspective, Communications in Computer and Information Science, Springer, Volume-628, pp. 104–110.**
20) Vonnegut S., (2016), Need-to-Know AppSec News Stories, Available at: https://www.checkmarx.com/2016/04/21/need-know-appsec-news-stories-april-2016/ Last Visit April 2018.
21) Plans B. E. A., (2014), Assessing Security and Privacy Controls in Federal Information Systems and Organizations, NIST Special Publication, 800, 53A.
22) Chouhan P. K., Yao F., Yerima S. Y., Sezer S., (2015), Software as a Service: Analyzing Security Issues, International Conference on Big Data and Analytics for Business, New Delhi, India, pp. 1-9.
23) The Sorry State of Software Security: Secure Development is Key, (2018) Available at: https://techbeacon.com/sorry-state-software-security-secure-development-key, Last Visit Oct 20 2018.
24) Is Your Security Up To Date? (2016) Available at: https://www.cisco.com/c/m/en_us/offers/sc04/2016-annual-security-report/index.html. Last Visit Nov 21 2018.
25) Secure SDLC Program Development/Enhancement, (2018) Available at: https://www.astechconsulting.com/software-development-life-cycle, Last Visit 15 May 2018.
26) Continuous Delivery: How to Have the Reliable Software Releases to Production at Any Time, (2018) Available at: http://www.softwaretestinghelp.com/what-is-continuous-delivery/, Last Visit March 31 2018.
27) Software Security, (2006) Available at: https://www.garymcgraw.com/technology/software-security/, Last Visit Nov 20 2018.
28) Risk Management Framework, (2015) Available at: https://www.us-cert.gov/bsi/articles/best-practices/risk-management/risk-management-framework-%28rmf%29c, Last Visit Nov 18 2018.
29) Secure Software Development Life Cycle Processes: A Technology Scouting Report, (2017), Available at https://securityintelligence.com/improve-application-security-immediately-with-these-5-software-development-practices/, Last Visit Sep 18 2018

30) Life without Computers, (2016), Available at: https://www.ukessays.com/essays/information-technology/life-without-computers.php, Last Visit Nov 20 2018.
31) Mikhailov L., (2003), Deriving Priorities from Fuzzy Pairwise Comparison Judgments, Fuzzy Sets and Systems, Volume 134, Issue 3, pp. 365-385.
32) Hahn W. J., Seaman S. L., Bikel R., (2012), Making Decisions With Multiple Attributes: A Case In Sustainability Planning, Graziadio Business Review, Volume 15, Issue 2, pp. 365-381.
33) Thirteen Principles to Ensure Enterprise System Security, (2013), Available at: https://searchsecurity.techtarget.com/opinion/Thirteen-principles-to-ensure-enterprise-system-security, Last Visit Oct 29 2018.
34) Colombo R. T., Pessôa M. S., Guerra A. C., Gomes C. C., (2012), Prioritization of Software Security Intangible Attributes, ACM SIGSOFT Software Engineering Notes, Volume 37, Issue 6, pp. 1-7.
35) Gulzar K., Sang J., Ramzan M., Kashif M., (2017), Fuzzy Approach to Prioritize Usability Requirements Conflicts: An Experimental Evaluation, IEEE Access, Volume 5, pp. 13570-13577.
36) Syamsuddin I., (2013), Multi Criteria Evaluation and Sensitivity Analysis on Information Security, International Journal of Computer Applications, Volume 69, Issue 24, pp. 22-25.
37) Zhu L., Aurum A., Gorton I., Jeffery R., (2005), Trade-off and Sensitivity Analysis in Software Architecture Evaluation using Analytic Hierarchy Process, Software Quality Journal, Volume 13, pp. 357-375.
38) Dlamini M. T., Eloff J. H., Eloff M. M., (2009), Information Security: The Moving Target, Computers & Security, Volume 28, Issue 3, pp. 189-198.
39) Cardenas A., Amin S., Sinopoli B., Giani A., Perrig A., Sastry S., (2009), Challenges for Securing Cyber Physical Systems, In Workshop on Future Directions in Cyber Physical Systems Security, Volume 5.
40) Parker D. B., (1992), Restating the Foundation of Information Security, Proceedings of the Eighth International Conference on Information Security, Netherlands, pp.139-151.
41) Meland P. H., Jensen J., (2008), Secure Software Design in Practice, In Availability, Reliability and Security, IEEE Access, pp. 1164-1171.
42) Chen C., Alfayez R., Srisopha K., Boehm B., Shi L., (2017), Why is It Important to Measure Maintainability and What are the Best Ways to Do It?", In Proceedings of the 39th International Conference on Software Engineering Companion, IEEE Press, pp. 377-378.
43) Van Der Linden D., Wupper H., (2010), A Method for Durability Analysis of Development Systems in Computing, Research Number 117 IK, pp. 1-17.
44) Cusick J. J., (2013), Durable Ideas in Software Engineering: Concepts, Methods and Approaches from My Virtual Toolbox, Bentham Science Publishers.
45) Gu L., Guo Y., Wang H., Zou Y. Z., Xie B., Shao W. Z., (2010), Runtime Software Trustworthiness Evidence Collection Mechanism based on TPM, Journal of Software, Volume 21, Issue 2, pp. 373-387.

46) Alarifi A., Alsaleh M., Alomar N., (2017), A Model for Evaluating the Security and Usability of e-Banking Platforms, Computing, Volume 99, Issue 5, pp. 519-535.
47) The Fundamental Tradeoffs, (2004), Available at: https://technet.microsoft.com/en-us/library/cc512573.aspx Last Visit Oct 23 2018.
48) The Critical Link in the Security Chain, (2015), Available at: https://www.vyapin.com/blog/the-critical-link-in-the-security-chain Last Visit Oct 29 2018.
49) Feenstra R. C., Knittel C. R., (2009), Re-assessing the US Quality Adjustment to Computer Prices: The Role of Durability and Changing Software, In Price Index Concepts and Measurement, University of Chicago Press, pp. 129-160.
50) Spacecraft Software Maintenance: An Effective Approach to Reducing Costs and Increasing Science Return, (1999), Available at: https://ntrs.nasa.gov/search.jsp?R=19990107384 Last Visit Nov 03 2018.
51) Secure Software Development Life Cycle Processes, (2013), Available at: https://www.us-cert.gov/bsi/articles/knowledge/sdlc-process/secure-software-development-life-cycle-processes Last Visit Nov 04 2018.
52) What is Confidentiality, Integrity, and Availability (CIA triad)?, (2015), Available at: https://whatis.techtarget.com/definition/Confidentiality-integrity-and-availability-CIA Last Visit Nov 04 2018.
53) What Is Computer Security?, (2003), Available at: https://flylib.com/books/en/2.514.1.15/1/, Last Visit Jan 04 2018.
54) Every Company Needs to Have a Security Program, (2008), Available at: https://www.appliedtrust.com/resources/security/every-company-needs-to-have-a-security-program, Last Visit Feb 06 2018.
55) Durable Cost Savings in Government IT, (2016), Available at: https://fcw.com/articles/2016/04/22/cost-savings-oped.aspx, Last Visit Oct 20 2018.
56) LeMay S. G., Benbrahim J., Chen X., (2007), Method and Apparatus for Software Authentication, U.S. Patent Number 7,201,662. Washington, DC: U.S. Patent and Trademark Office.
57) Zavadskas E. K., Govindan K., Antucheviciene J., Turskis Z., (2016), Hybrid Multiple Criteria Decision Making Methods: A Review of Applications for Sustainability Issues, Economic Research-Ekonomska Istraživanja, Volume 29, Issue 1, pp. 857-887.
58) Goals of Security Confidentiality, Integrity, and Availability, (2018), Available at: https://www.examcollection.com/certification-training/security-plus-goals-of-security-confidentiality-integrity-availability.html Last Visit Oct 20 2018.
59) Thomas R., (1994), Durable, Low Cost Educational Software, In Computer Assisted Learning: Selected Contributions from the CAL'93 Symposium, pp. 65-72.
60) Mougouei D., (2017), PAPS: A Scalable Framework for Prioritization and Partial Selection of Security Requirements, Cornell University Library, Publication Number - eprint arXiv: 170600166.
61) Evans R., (2015), Integrating Security in to the Undergraduate Software Engineering Curriculum, UN F The sesand Dissertations, 600.

62) Hoehl M., (2013), Framework for Building a Comprehensive Enterprise Security Patch Management Program, STI Graduate Student Research, SANS.
63) Chatterjee K., Gupta D., De A., (2013), A Framework for Development of Secure Software, CSI Transactions on ICT, Volume 1, Issue 2, pp. 143-157.
64) Khan S. A., Khan R. A., (2012), A Framework to Quantify Security: Complexity Perspective, International Journal of Information and Education Technology, Volume 2, Issue 5, pp. 439.
65) Agrawal A., Khan R. A., (2011), A Framework for Vulnerability Minimization: Object Oriented Design Perspective, Computer and Communication Technology, pp. 499-504.
66) Jain S., Ingle M., (2011), Review of Security Metrics in Software Development Process, International Journal of Computer Science and Information Technologies, Volume 2, Issue 6, pp. 2627-2631.
67) Security Awareness Program Special Interest Group PCI Security Standards Council, (2014), Information Supplement: Best Practices for Implementing a Security Awareness Program, PCI Data Security Standard, Version 1.
68) Bartlett E. V., Simpson S., (2013), Durability and Reliability, Alternative Approaches to Assessment of Component Performance over Time, Available at: https://www.irbnet.de/daten/iconda/CIB8616.pdf, Last Visit Sep 20 2018.
69) Hayden E., Assante M., Conway T., (2014), An Abbreviated History of Automation & Industrial Controls Systems and Cyber Security, A SANS Analyst Whitepaper.
70) Hneif M., Lee S. P., (2011), Using Guidelines to Improve Quality in Software Non Functional Attributes, IEEE Software, Volume 28, Issue 6, pp. 72-77.
71) Vandegriend B., (2006), How to Create Maintainable Software, Available at: http://www.basilv.com/psd/blog/2006/the-importance-of-maintainable-software. Last Visit Sep 25 2018.
72) Seong N. H., (2012). A Reliable, Secure Phase Change Memory as a Main Memory, PhD Thesis, Georgia Institute of Technology.
73) Kaufman L. M., (2009), Data Security in the World of Cloud Computing, IEEE Security & Privacy, Volume 7, Issue 4, pp. 54-57.
74) Kinney T., Reppen D., Yee D., Schlatter J., (2002), Reliability, Availability, Maintainability, Durability (RAMD) Testing and Control System Development, Final Report-DOE/CH/10941-2, Catalytica Combustion Systems, Mountain View, CA (US).
75) Afrin A., Sadiq M., (2017), An Integrated Approach for the Selection of Software Requirements using Fuzzy AHP and Fuzzy TOPSIS Method, In Intelligent Computing, Instrumentation and Control Technologies, IEEE Press, pp. 1094-1100.
76) Chong C. Y., Lee S. P., Ling T. C., (2014), Prioritizing and Fulfilling Quality Attributes for Virtual Lab Development through Application of Fuzzy Analytic Hierarchy Process and Software Development Guidelines, Malaysian Journal of Computer Science, Volume 27, Issue 1, pp. 1-19.
77) Goli D., (2013), Group Fuzzy TOPSIS Methodology in Computer Security Software Selection, International Journal of Fuzzy Logic Systems, Volume 3, Issue 2, pp. 29-47.

78) Dubey S. K., Singh A., (2013), Evaluation of Usability using Soft Computing Technique, International Journal of Science Engineering and Research, Volume 4, Issue 12, pp. 162-166.
79) Shi L., Yang S., Li K., Yu B. G., (2012), Developing an Evaluation Approach for Software Trustworthiness using Combination Weights and TOPSIS, Journal of Software, Volume 7, Issue 3, pp. 532-543.
80) FadlElssied N. O., Ibrahim, O., (2011), A Review of Fuzzy Mechanisms for E-government Security, International Journal of Computer Applications, Volume 34, Issue 7, pp. 16-22.
81) Özdağoğlu A., Özdağoğlu G., (2007), Comparison of AHP and Fuzzy AHP for the Multi-Criteria Decision Making Processes with Linguistic Evaluations, İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, Volume 6, Issue 11, pp. 65-85.
82) Van Laarhoven P. J. M., Pedrycz W., (1983), A Fuzzy Extension of Saaty's Priority Theory, Fuzzy Sets and Systems, Volume 11, Issue 3, pp. 229-241.
83) Develop Zone, (2016), Available at: https://dzone.com/articles/the-ultimate-list-of-100-software-testing-quotes-2, Last Visit Jan 24 2018.
84) Jansson A. S., (2007), Software Maintenance and Process Improvement, CMMI, UPTEC STS07037.
85) Tekinaslan H., (2018), Available at: https://twitter.com/htkaslan Last Visit Jan 24 2018.
86) Professional Software Development, (2006) Available at: http://www.basilv.com/psd/blog/2006/how-to-create-maintainable-software Last Visit Jan 30 2018.
87) Martin R. C., (2009), Clean Code: A Handbook of Agile Software Craftsmanship, Pearson Education Press.
88) Mitnick K., (2000), Available at: https://www.theregister.co.uk/2000/03/02/kevin_mitnick_was_no_hacker/ Last Visit Jan 24 2018.
89) Schneier B., (2009), Schneier on Security, John Wiley & Sons, pp. 1-2.
90) Coders Trust, (2018), Available at: https://coderstrust.tumblr.com/post/116455270040/top-20-coding-quotes Last Visit Jan 24 2018.
91) Implementing System Quality Attributes, (2007), Available at: https://msdn.microsoft.com/en-us/library/bb402962.aspx Last Visit Jan 30 2018.
92) Sutherland J., (1995), Business Objects in Corporate Information Systems, ACM Computing Surveys, Volume 27, Issue 2, pp. 274-276.
93) Sommardahl B, Durable Software, (2013) Awkward Coder Learning to Behave in Public, pp. 5-8.
94) Abdulrazeg A. A., Norwawi N. M., Basir N., (2012), Security Measurement based on GQM to Improve Application Security During Requirements Stage, International Journal of Cyber Security and Digital Forensics, Volume 1, Issue 3, pp. 211-220.
95) **Kumar R., Khan S. A., Agrawal A., Khan R. A., (2018), Measuring the Security Attributes through Fuzzy Analytic Hierarchy Process: Durability Perspective, ICIC**

Express Letters-An International Journal of Research and Surveys, Volume 12, Number 6, pp. 615-620.

96) Crofts K., Bisman J., (2010), Interrogating Accountability: An Illustration of the Use of Leximancer Software for Qualitative Data Analysis, Qualitative Research in Accounting & Management, Volume 7, Issue 2, pp. 180-207.

97) Baas S. M., Kwakernaak H., (1977), Rating and Ranking of Multiple - Aspect Alternatives Using Fuzzy Sets, Automatica, Volume 13, Number 1, pp.47-58.

98) Kluwer W., (2015), Starting your Software Security Assurance Program, ITARC, Stockholm, Sweden, May 21, 2015.

99) Kharat M. G., Kamble S. J., Raut R. D., Kamble S. S., (2016), Identification and Evaluation of Landfill Site Selection Criteria using a Hybrid Fuzzy Delphi, Fuzzy AHP and DEMATEL based Approach, Modeling Earth Systems and Environment, Volume 2, Issue 2, pp. 98.

100) Saaty T. L., (1995), Transport Planning with Multiple Criteria: The Analytic Hierarchy Process Applications and Progress Review, Journal of Advanced Transportation, Volume 29, Issue 1, pp. 81-126.

101) Zadeh L.A., (1965), Fuzzy Sets, Information and Control, Volume 8, Issue 3, pp. 338–353.

102) Gohar A. S., Khanzadi M., Farmani M., (2012), Identifying and Evaluating Risks of Construction Projects in Fuzzy Environment: A Case Study in Iranian Construction Industry, Indian Journal of Science and Technology, Volume 5, Issue 11, pp. 15-25.

103) Millet I., Saaty T. L., (2000), On the Relativity of Relative Measures–Accommodating both Rank Preservation and Rank Reversals in the AHP, European Journal of Operational Research, Volume 121, Issue 1, pp. 205-212.

104) Saaty T. L., (2008), The Analytic Hierarchy Process, McGraw-Hill, New York.

105) Csutora R., Buckley J. J., (2001), Fuzzy Hierarchical Analysis: The Lambda-Max Method, Fuzzy Sets and Systems, pp. 181-195.

106) Ammar S., Wright R., (2000), Applying Fuzzy Set Theory to Performance Evaluation, Socio-Economic Planning Sciences, Volume 34, pp. 285-302.

107) Mishra A., Dubey S. K., (2014), Evaluation of Reliability of Object Oriented Software System Using Fuzzy Approach, In Confluence The Next Generation Information Technology Summit, pp. 806-809.

108) Chang C. W., Wu C. R., Lin H. L., (2008), Integrating Fuzzy Theory and Hierarchy Concepts to Evaluate Software Quality, Software Quality Journal, Volume 16, Number 2, pp. 263-276.

109) Srivastava P. R., Singh A. P., Vageesh K.V., (2010), 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.

110) Kumar R., Khan S. A., Agrawal A., Khan R. A., (2018), Security Durability Assessment Framework, Indian Patent Number 201711032601, Patent and Trademark Office, India.

111) Addressing Software Security in Federal Acquisition Process, (2011) Available at: http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=55E94ECFDC445D24E058F6334BB525CD?doi=10.1.1.300.2941&rep=rep1&type=pdf, Last visit 23 July 2018.
112) Bstieler L., (2005), The Moderating Effect of Environmental Uncertainty on New Product Development and Time Efficiency, Journal of Product Innovation Management, Volume 22, Issue 3, pp. 267-284.
113) Triantaphyllou E., Mann S. H., (1995), Using the Analytic Hierarchy Process for Decision Making in Engineering Applications: Some Challenges, International Journal of Industrial Engineering: Applications and Practice, Volume 2, Issue 1, pp. 35-44.
114) Kumar R., Khan S. A., Agrawal A., Khan R. A., (2018), Durable Security Assessment through Fuzzy Multi Criteria Decision Technique, Malaysian Journal of Computer Science, Under Review from Sep 2017.
115) Chowdhury I., Zulkernine M., (2010), Can Complexity, Coupling, and Cohesion Metrics be Used as Early Indicators of Vulnerabilities?, In Proceedings of the 2010 ACM Symposium on Applied Computing, pp. 1963-1969.
116) Abbadi Z., (2011), Security Metrics What Can We Measure?, In Open Web Application Security Project (OWASP), Nova Chapter Meeting Presentation on Security Metrics, Volume 2.
117) Siddiqui S. T., (2017), Significance of Security Metrics in Secure Software Development, International Journal of Applied Information Systems, Volume 12, Issue 6, pp. 10-15.
118) Yadav S., Sunil S., Uttpal S., (2014), A Review of Object-Oriented Coupling and Cohesion Metrics, International Journal of Computer Science Trends and Technology, Volume 2, Issue 5, pp. 45-55.
119) Mohammed O. S., Taha D. B., (2016), Conducting Multi-Class Security Metrics from Enterprise Architect Class Diagram, International Journal of Computer Science and Information Security, Volume 14, Issue 4, pp. 56.
120) Alshammari B. M., (2011), Quality Metrics for Assessing Security Critical Computer Programs, PhD Thesis, Queensland University of Technology.
121) Krishna G., Joshi R. K., (2010), Inheritance Metrics: What Do They Measure?, In Proceedings of the 4th Workshop on Mechanisms for Specialization, Generalization and inheritance, ACM, p. 1.
122) Kumar M. D. S., Prasad R. S., (2015), New Metrics for System Understandability of Inheritance Hierarchies, International Journal of Research Studies in Computer Science and Engineering, Volume 2, Issue 3, pp. 59-62.
123) Peterson R. S., Wong B., Sirer E. G., (2011), A Content Propagation Metric for Efficient Content Distribution, In ACM SIGCOMM Computer Communication Review, Volume 41, Issue 4, pp. 326-337.
124) Taha D. B., Mohammed O. S., (2017), Conducting Security Metrics for Object-Oriented Class Design, International Journal of Computer Science and Information Security, Volume 15, Issue 8, pp. 20-27.
125) Agrawal A., Khan R. A., (2014), Assessing Impact of Cohesion on Security: An Object Oriented Design Perspective. Pensee Journal, Volume 76, Issue 2, pp. 45-54.

126) Alvaro A., Almeida E. S., Meira S. R. L., (2005), Towards a Software Component Quality Model, In Submitted to the 5th International Conference on Quality Software, pp. 103-116.
127) De Boer R. C., Van Vliet H., (2008), Architectural Knowledge Discovery with Latent Semantic Analysis: Constructing a Reading Guide for Software Product Audits, Journal of Systems and Software, Volume 81, Issue 9, pp. 1456-1469.
128) Selic B., (2007), From Model Driven Development to Model Driven Engineering, Euromicro Technical Committee on Real-Time Systems, pp. 3-10.
129) Lee M. C., (2014), Software Quality Factors and Software Quality Metrics to Enhance Software Quality Assurance, British Journal of Applied Science & Technology, Volume 4, Issue 21, pp. 3069-3095.
130) Vinte C., (2012), Software Architecture Coupling Metric for Assessing Operational Responsiveness of Trading Systems, Informatica Economica, Volume 16, Issue 4, pp. 105-112.
131) McNaughton M., Baker C. R., Galatali T., Salesky B., Urmson C., Ziglar J., (2008), Software Infrastructure for an Autonomous Ground Vehicle, Journal of Aerospace Computing, Information, and Communication, Volume 5, Issue 12, pp. 491-505.
132) Rating Definition by Oxford Dictionaries, (2018), Available at: https://en.oxforddictionaries.com/definition/rating, Last Visit Oct 25 2018.
133) How Compliance and Security Requirements May Conflict, (2008), Available at: https://technet.microsoft.com/en-us/library/2008.06.desktopfiles.aspx, Last Visit Sep 20 2018.
134) Syau Y. H., Hsieh H.T., Lee E. S., (2001), Fuzzy Numbers in the Credit Rating of Enterprise Financial Condition, Review of Quantitative Finance and Accounting, Volume 17, Issue 4, pp. 351–360.
135) Population Parameter, (2018), Available at: https://math.tutorvista.com/statistics/population-parameter.html. Last Visit Sep 30 2018.
136) Business Statistics, (2004), Available at: https://learn.saylor.org/course/bus204, Last Visit Aug 30 2018.
137) Scalability Testing: Complete Guide, (2014), Available at: https://www.guru99.com/scalability-testing.html, Last Visit July 30 2018.
138) Briand L., Emam K. E., Moraska S., (1995), Theoretical and Empirical Validation of Software Product Metrics, Technical Report Number- ISERN-95-03, International Software Engineering Research Network, Version 1.
139) Zelkowitz M. V., Wallace D., (1977), Experimental Validation in Software Engineering, Information and Software Technology, Volume 39, Issue 11, pp. 735-743.
140) Support or Reject Null Hypothesis in Easy Steps, (2009), Available at: http://www.statisticshowto.com/support-or-reject-null-hypothesis/, Last Visit Nov 18 2018.
141) Kumar R., Khan S. A., Agrawal A., Khan R. A., (2014), Revisiting Software Security Risks, British Journal of Mathematics & Computer Science, Volume 11, Issue 6, pp. 1-10.

142) Kumar R., Khan S. A., Agrawal A., Khan R. A., (2015), Managing Software Security Risk: Design Perspective Indian Patent Number 1781/DEL/2015, Patent and Trademark Office, India.

143) Kumar R., Khan S. A., Khan R. A., (2015) Durable Security in Software Development: Needs and Importance, CSI Communication, pp. 34-36, Oct 2015.

144) Kumar R., Khan S. A., Agrawal A., Khan R. A., (2017), Fuzzy Analytic Hierarchy Process for Software Durability: Security Risks Perspective, Advances in Intelligent Systems and Computing (Originally Published with the Title : Advances in Intelligent and Soft Computing), Volume-508, Springer, pp. 613-620.

145) Kaur J., Kumar R., Khan S. A., Agrawal A., Khan R. A., Choudhary R. K., (2019), Adaptive Neuro-Fuzzy Inference System for Security Risk Assessment: A Design Perspective, Frontiers of Information Technology & Electronic Engineering, Springer, 2019. (IF: 0.622) (Accepted)

146) Kumar R., Khan S. A., Khan R. A., (2016) Durability Challenges in Software Engineering, Crosstalk-The Journal of Defense Software Engineering, Volume 8, pp. 29-31.

147) Kumar R., Khan S. A., Khan R. A., (2016), Modern Security Challenges, International Journal of Innovations & Advancement in Computer Science, Volume 5, Issue 4 pp. 67-72.

148) Kumar R., Kapil G., Khan S. A., (2016), Basic Concepts of Durable Software, Proceeding of the National Conference on Information Security Challenges (NCISC-2016), Organized by Department of Information Technology, Babasaheb Bhimrao Ambedkar University, Lucknow, India, February 2016.

149) Kumar R., Khan S. A., Agrawal A., Khan R. A., (2018), Security Assessment through Fuzzy Delphi Analytic Hierarchy Process, ICIC Express Letters-An International Journal of Research and Surveys, Volume 12, Number 10, pp. 56-62.

150) Kumar R., Khan S. A., Khan R. A., (2016) Analytical Network Process for Software Security: A Design Perspective, CSI Transactions on ICT, Springer, pp. 1-4, 2016.

151) Kumar R., Khan S. A., Agrawal A., Khan R. A., Durable and Secure Software, National Academy Science Letters, Under Review from 31 May 2017.

152) Kumar R., Khan S. A., Agrawal A., Khan R. A., Multi-level Fuzzy System for Security Assessment: Durability Perspective, International Journal of Innovative Computing, Information and Control, Under Review from 14 Jan 2018.

Appendix A

Compiled Comments from Reviewers

"…This work proposes a quantitative methodology for making security design decisions of software project. The central term "software security durability" is not a well-established concept. The listed factors are too small. The literature review does not help me to understand what are software security and software durability. The authors should position the work in a big picture of security software engineering. The practicality of the proposal needs to be justified with real-world cases. It seems that the author utilizes weights to quantify influences of different criterions to security durability evaluation. What kind of methods are used for calculate these weights, subjective weighting or objective weighting?….." ***Prof. Elisa Bertino, < bertino@purdue.edu>***
"……..The AHP was a known algorithm; no improvement of the algorithm was done. Application of new method/process was informed but the validation of the process and result was not clearly explained. Authors need to clearly distinguish between durability and security…..…….." ***Dr. Bassam El Ali, < belali@kfupm.edu.sa>***
"…….Potential contributions could be the main advantages of adopting durability in security development, but it is uncompleted. Most of the text is the description of security and durability issues. Although some parts provide quite good overview of issues mentioned in the literature; challenges or rather requirements for a solid secure software development process, generally the text is large extent stating elementary facts, obvious to anyone from security domain……..." ***Dr. Hana Chockler, < hana.chockleratkcl.ac.uk>***
"……This work talks about another application of fuzzy-AHP - a decision making technique. Fuzzy AHP and priority assignment seems to be a well-researched topic and easily understandable content is available on the Internet. This work defines a particular criteria to be used for a specific goal. i.e attributes of security durability trustworthiness, dependability and human trust are chosen as criteria to make decisions for durable software design. But authors are not cleared about the threshold value. In this work, for defuzzification, authors are taken threshold value (alpha=0.5 and beta=0.5), authors do not provide the variation of threshold values in the results…….……..…..." ***Prof. Manjusha Pandey, <manjushafcs@kiit.ac.in>***

Appendix B

Questionnaire Form for Evaluating the Importance of Security Durability Attributes

Details and Description: Due to the heavy cost incurred on software development and maintenance, secure software with longer service life span is in high demand. This property of software is termed as security durability. During software development, security durability of software can be improved by improving its attributes. Hence the organizations need to identify, correlate, estimate and improve security durability attributes during software development. To help developers, the researcher has proposed a methodology to evaluate and improve security durability of software during its development. The methodology is based upon Multi Criteria Decision Analysis Methods.

Your suggestions will surely help to improve the methodology. So you are requested to kindly give your opinion for the given set of questionnaire. The scale for answers has been given in the table 1. The responses are to be given in numeric form. The reciprocal numeric values represent the opposite of the importance level.

Table 1: Scale of Linguistic Values with Numerical Values

S. No.	Linguistic Values	Numeric Values	Reciprocal Values
1	Equal Important (Eq)	1	1
2	Intermediate Value between Equal and Weekly (E & W)	2	2^{-1}
3	Weekly Important (WI)	3	3^{-1}
4	Intermediate Value between Weekly and Essential (W & E)	4	4^{-1}
5	Essential Important (EI)	5	5^{-1}
6	Intermediate Value between Essential and Very Strongly (E & VS)	6	6^{-1}
7	Very Strongly Important (VS)	7	7^{-1}
8	Intermediate Value between Very Strongly and Extremely (VS & ES)	8	8^{-1}
9	Extremely Important (ES)	9	9^{-1}

Please read the following questions and put check marks on the pair wise comparison matrices. If a criteria on the left is more important than the matching one on the right, put your check mark to the left of the importance ‘‘Equal (1)’’ under the importance level you prefer. If a criteria on the left is less important than the matching one on the right, put your check mark to the right of the importance ‘Equal (1)’ under the importance level you. Reciprocal value means the opposite effect of the factor of assigned value. Here, total eleven groups are available. Please put your mark for each group.

A. With respect to the criteria “Security Durability”

Question 1: How important is the relation between “Dependability (C1)” and “Trustworthiness (C2)”?
Question 2: How important is the relation between “Dependability (C1)” and “Human Trust (C3)”?
Question 3: How important is the relation between “Trustworthiness (C2)” and “Human Trust (C3)”?

Importance of One Criteria Over Another																			
Q.N.		**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	$\mathbf{2^{-1}}$	$\mathbf{3^{-1}}$	$\mathbf{4^{-1}}$	$\mathbf{5^{-1}}$	$\mathbf{6^{-1}}$	$\mathbf{7^{-1}}$	$\mathbf{8^{-1}}$	$\mathbf{9^{-1}}$	
1	C1																		C2
2	C1																		C3
3	C2																		C3

B. With respect to the criteria "Dependability (C1)"

Question 1: How important is the relation between "Availability (C11)" and "Reliability (C12)"?
Question 2: How important is the relation between "Availability (C11)" and "Maintainability (C13)"?
Question 3: How important is the relation between "Availability (C11)" and "Confidentiality (C14)"?
Question 4: How important is the relation between "Availability (C11)" and "Authentication (C15)"?
Question 5: How important is the relation between "Reliability (C12)" and "Maintainability (C13)"?
Question 6: How important is the relation between "Reliability (C12)" and "Confidentiality C14)"?
Question 7: How important is the relation between "Reliability (C12)" and "Authentication (C15)"?
Question 8: How important is the relation between "Maintainability (C13)" and "Confidentiality C14)"?
Question 9: How important is the relation between "Maintainability (C13)" and "Authentication (C15)"?
Question 10: How important is the relation between "Confidentiality C14)" and "Authentication (C15)"?

Importance of One Criteria Over Another																		
Q.N.		**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	2^{-1}	3^{-1}	4^{-1}	5^{-1}	6^{-1}	7^{-1}	8^{-1}	9^{-1}	
1	C11																	C12
2	C11																	C13
3	C11																	C14
4	C11																	C15
5	C12																	C13
6	C12																	C14
7	C12																	C15
8	C13																	C14
9	C13																	C15
10	C14																	C15

C. With respect to the criteria "Trustworthiness (C2)"

Question 1: How important is the relation between "Availability (C21)" and "Reliability (C22)"?
Question 2: How important is the relation between "Availability (C21)" and "Maintainability (C23)"?
Question 3: How important is the relation between "Availability (C21)" and "Accountability (C24)"?
Question 4: How important is the relation between "Availability (C21)" and "Survivability (C25)"?
Question 5: How important is the relation between "Reliability (C22)" and "Maintainability (C23)"?
Question 6: How important is the relation between "Reliability (C22)" and "Accountability (C24)"?
Question 7: How important is the relation between "Reliability (C22)" and "Survivability (C25)"?
Question 8: How important is the relation between "Maintainability (C23)" and "Accountability (C24)"?
Question 9: How important is the relation between "Maintainability (C23)" and "Survivability (C25)"?
Question 10: How important is the relation between "Accountability (C24)" and "Survivability (C25)"?

Importance of One Criteria Over Another																		
Q.N.		**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	2^{-1}	3^{-1}	4^{-1}	5^{-1}	6^{-1}	7^{-1}	8^{-1}	9^{-1}	
1	C21																	C22
2	C21																	C23
3	C21																	C24
4	C21																	C25
5	C22																	C23
6	C22																	C24
7	C22																	C25
8	C23																	C24
9	C23																	C25
10	C24																	C25

D. With respect to the criteria "Human Trust (C3)"

Question 1: How important is the relation between "Reliability (C31)" and "Consumer Integrity (C32)"?
Question 2: How important is the relation between "Reliability (C31)" and "Accountability (C33)"?
Question 3: How important is the relation between "Reliability (C31)" and "Confidentiality (C34)"?
Question 4: How important is the relation between "Reliability (C31)" and "Authentication (C35)"?
Question 5: How important is the relation between "Consumer Integrity (C32)" and "Accountability (C33)"?
Question 6: How important is the relation between "Consumer Integrity (C32)" and "Confidentiality (C34)"?
Question 7: How important is the relation between "Consumer Integrity (C32)" and "Authentication"?
Question 8: How important is the relation between "Accountability (C33)" and "Confidentiality (C34)"?
Question 9: How important is the relation between "Accountability (C33)" and "Authentication (C35)"?
Question 10: How important is the relation between "Confidentiality (C34)" and "Authentication (C35)"?

Importance of One Criteria Over Another																		
Q.N.		**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	2^{-1}	3^{-1}	4^{-1}	5^{-1}	6^{-1}	7^{-1}	8^{-1}	9^{-1}	
1	C31																	C32
2	C31																	C33
3	C31																	C34
4	C31																	C35
5	C32																	C33
6	C32																	C34
7	C32																	C35
8	C33																	C34
9	C33																	C35
10	C34																	C35

E. With respect to the criteria "Availability (C11)"

Question 1: How important is the relation between "Auditability (C111)" and "Feasibility (C112)"?
Question 2: How important is the relation between "Auditability (C111)" and "Accessibility (C113)"?
Question 3: How important is the relation between "Auditability (C111)" and "Software Effectiveness Evaluation (C114)"?
Question 4: How important is the relation between "Auditability (C111)" and "Operational Controls (C115)"?
Question 5: How important is the relation between "Feasibility (C112)" and "Accessibility (C113)"?
Question 6: How important is the relation between "Feasibility (C112)" and "Software Effectiveness Evaluation (C114)"?
Question 7: How important is the relation between "Feasibility (C112)" and "Operational Controls (C115)"?
Question 8: How important is the relation between "Accessibility (C113)" and "Software Effectiveness Evaluation (C114)"?
Question 9: How important is the relation between "Accessibility (C113)" and "Operational Controls (C115)"?
Question 10: How important is the relation between "Software Effectiveness Evaluation (C114)" and "Operational Controls (C115)"?

Importance of One Criteria Over Another																		
Q.N.		**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	2^{-1}	3^{-1}	4^{-1}	5^{-1}	6^{-1}	7^{-1}	8^{-1}	9^{-1}	
1	C111																	C112
2	C111																	C113
3	C111																	C114
4	C111																	C115
5	C112																	C113
6	C112																	C114
7	C112																	C115
8	C113																	C114
9	C113																	C115
10	C114																	C115

F. With respect to the criteria "Reliability (C12)"

Question 1: How important is the relation between "Feasibility (C121)" and "Time-efficiency (C122)"?
Question 2: How important is the relation between "Feasibility (C121)" and "User Satisfaction (C123)"?
Question 3: How important is the relation between "Feasibility (C121)" and "Business Continuity (C124)"?
Question 4: How important is the relation between "Time-efficiency (C122)" and "User Satisfaction (C123)"?
Question 5: How important is the relation between "Time-efficiency (C122)" and "Business Continuity (C124)"?
Question 6: How important is the relation between "User Satisfaction (C123)" and "Business Continuity (C124)"?

Importance of One Criteria Over Another																		
Q.N.		**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	2^{-1}	3^{-1}	4^{-1}	5^{-1}	6^{-1}	7^{-1}	8^{-1}	9^{-1}	
1	C121																	C122
2	C121																	C123
3	C121																	C124
4	C122																	C123
5	C122																	C124
6	C123																	C124

G. With respect to the criteria "Maintainability (C13)"

Question 1: How important is the relation between "Auditability (C131)" and "Scalability (C132)"?
Question 2: How important is the relation between "Auditability (C131)" and "Traceability (C133)"?
Question 3: How important is the relation between "Auditability (C131)" and "Detectability (C134)"?
Question 4: How important is the relation between "Auditability (C131)" and "Extensibility (C135)"?
Question 5: How important is the relation between "Auditability (C131)" and "Flexibility (C136)"?
Question 6: How important is the relation between "Auditability (C131)" and "Accessibility (C137)"?
Question 7: How important is the relation between "Auditability (C131)" and "Time-efficiency (C138)"?
Question 8: How important is the relation between "Scalability (C132)" and "Traceability (C133)"?
Question 9: How important is the relation between "Scalability (C132)" and "Detectability (C134)"?
Question 10: How important is the relation between "Scalability (C132)" and "Extensibility (C135)"?
Question 11: How important is the relation between "Scalability (C132)" and "Flexibility (C136)"?
Question 12: How important is the relation between "Scalability (C132)" and "Accessibility (C137)"?
Question 13: How important is the relation between "Scalability (C132)" and "Time-efficiency (C138)"?
Question 14: How important is the relation between "Traceability (C133)" and "Detectability (C134)"?
Question 15: How important is the relation between "Traceability (C133)" and "Extensibility (C135)"?
Question 16: How important is the relation between "Traceability (C133)" and "Flexibility (C136)"?
Question 17: How important is the relation between "Traceability (C133)" and "Accessibility (C137)"?
Question 18: How important is the relation between "Traceability (C133)" and "Time-efficiency (C138)"?
Question 19: How important is the relation between "Detectability (C134)" and "Extensibility (C135)"?
Question 20: How important is the relation between "Detectability (C134)" and "Flexibility (C136)"?
Question 21: How important is the relation between "Detectability (C134)" and "Accessibility (C137)"?
Question 22: How important is the relation between "Detectability (C134)" and "Time-efficiency (C138)"?
Question 23: How important is the relation between "Extensibility" and "Flexibility (C136)"?
Question 24: How important is the relation between "Extensibility (C135)" and "Accessibility (C137)"?
Question 25: How important is the relation between "Extensibility (C135)" and "Time-efficiency (C138)"?
Question 26: How important is the relation between "Flexibility (C136)" and "Accessibility (C137)"?
Question 27: How important is the relation between "Flexibility (C136)" and "Time-efficiency (C138)"?
Question 28: How important is the relation between "Accessibility (C137)" and "Time-efficiency (C138)"?

Importance of One Criteria Over Another																			
Q.N.		**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	2^{-1}	3^{-1}	4^{-1}	5^{-1}	6^{-1}	7^{-1}	8^{-1}	9^{-1}	
1	C131																		C132
2	C131																		C133
3	C131																		C134
4	C131																		C135
5	C131																		C136
6	C131																		C137
7	C131																		C138
8	C132																		C133
9	C132																		C134
10	C132																		C135
11	C132																		C136
12	C132																		C137
13	C132																		C138
14	C133																		C134
15	C133																		C135
16	C133																		C136
17	C133																		C137
18	C133																		C138
19	C134																		C135
20	C134																		C136
21	C134																		C137
22	C134																		C138
23	C135																		C136
24	C135																		C137
25	C135																		C138
26	C136																		C137
27	C136																		C138
28	C137																		C138

H. With respect to the criteria "Confidentiality (C14)"

Question 1: How important is the relation between"User Satisfaction (C141)" and "Software Effectiveness Evaluation (C142)"?

Question 2: How important is the relation between"User Satisfaction (C141)" and 'Operational Controls (C143)"?

Question 3: How important is the relation between"Software Effectiveness Evaluation (C142)" and 'Operational Controls (C143)"?

Importance of One Criteria Over Another																			
Q.N.		**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	2^{-1}	3^{-1}	4^{-1}	5^{-1}	6^{-1}	7^{-1}	8^{-1}	9^{-1}	
1	C141																		C142
2	C141																		C143
3	C142																		C143

I. With respect to the criteria "Authentication (C15)"

Question 1: How important is the relation between "Psychological Acceptability (C151)" and "User Satisfaction (C152)"?

Question 2: How important is the relation between "Psychological Acceptability (C151)" and "Software Effectiveness Evaluation (C153)"?

Question 3: How important is the relation between "Psychological Acceptability (C151)" and "Operational Controls (C154)"?

Question 4: How important is the relation between "User Satisfaction (C152)" and "Software Effectiveness Evaluation (C153)"?

Question 5: How important is the relation between "User Satisfaction (C152)" and "Operational Controls (C154)"?

Question 6: How important is the relation between "Software Effectiveness Evaluation (C153)" and "Operational Controls (C154)"?

Importance of One Criteria Over Another																		
Q.N.		**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	2^{-1}	3^{-1}	4^{-1}	5^{-1}	6^{-1}	7^{-1}	8^{-1}	9^{-1}	
1	C151																	C152
2	C151																	C153
3	C151																	C154
4	C152																	C153
5	C152																	C154
6	C153																	C154

J. With respect to the criteria "Survivability (C25)"

Question 1: How important is the relation between"Detectability (C251)" and "Extensibility (C252)"?

Question 2: How important is the relation between"Detectability (C251)" and 'Flexibility (C253)"?

Question 3: How important is the relation between"Extensibility (C252)" and 'Flexibility (C253)"?

Importance of One Criteria Over Another																		
Q.N.		**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	2^{-1}	3^{-1}	4^{-1}	5^{-1}	6^{-1}	7^{-1}	8^{-1}	9^{-1}	
1	C141																	C142
2	C141																	C143
3	C142																	C143

K. With respect to the criteria "Authentication (C32)"

Question 1: How important is the relation between "Psychological Acceptability (C321)" and "User Satisfaction (C322)"?

Question 2: How important is the relation between "Psychological Acceptability (C321)" and "Business Continuity (C323)"?

Question 3: How important is the relation between "Psychological Acceptability (C321)" and "Operational Controls (C324)"?

Question 4: How important is the relation between "User Satisfaction (C322)" and "Business Continuity (C323)"?

Question 5: How important is the relation between "User Satisfaction (C322)" and "Operational Controls (C324)"?

Question 6: How important is the relation between "Business Continuity (C323)" and "Operational Controls (C324)"?

Importance of One Criteria Over Another																		
Q.N.		**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	2^{-1}	3^{-1}	4^{-1}	5^{-1}	6^{-1}	7^{-1}	8^{-1}	9^{-1}	
1	C321																	C322
2	C321																	C323
3	C321																	C324
4	C322																	C323
5	C322																	C324
6	C323																	C324

Your Comments (Please mark corrections as and where required): Please find details in E-Mail:

Expert's Name and Signature:
(Please return this to: Rajeev Kumar (rs0414@gmail.com), D/O Information Technology, SIST, Babasaheb Bhimrao Ambedkar University, Lucknow, Uttar Pradesh, India)

Appendix: C
Sample: Questionnaire Reports

Questionnaire Form for Evaluating the Importance of Security Durability Attributes

Details and Description: Due to the heavy cost incurred on software development and maintenance, secure software with longer service life span is in high demand. This property of software is termed as security durability. During software development, security durability of software can be improved by improving its attributes. Hence the organizations need to identify, correlate, estimate and improve security durability attributes during software development. To help developers, the researcher has proposed a methodology to evaluate and improve security durability of software during its development. The methodology is based upon Multi Criteria Decision Analysis Methods.

Your suggestions will surely help to improve the methodology. So you are requested to kindly give your opinion for the given set of questionnaire. The scale for answers has been given in the table 1. The responses are to be given in numeric form. The reciprocal numeric values represent the opposite of the importance level.

Table 1: Scale of Linguistic Values with Numerical Values

S. No.	Linguistic Values	Numeric Values	Reciprocal Values
1	Equal Important (Eq)	1	1
2	Intermediate Value between Equal and Weekly (E & W)	2	2^{-1}
3	Weekly Important (WI)	3	3^{-1}
4	Intermediate Value between Weekly and Essential (W & E)	4	4^{-1}
5	Essential Important (EI)	5	5^{-1}
6	Intermediate Value between Essential and Very Strongly (E & VS)	6	6^{-1}
7	Very Strongly Important (VS)	7	7^{-1}
8	Intermediate Value between Very Strongly and Extremely (VS & ES)	8	8^{-1}
9	Extremely Important (ES)	9	9^{-1}

Please read the following questions and put check marks on the pair wise comparison matrices. If a criteria on the left is more important than the matching one on the right, put your check mark to the left of the importance ''Equal (1)'' under the importance level you prefer. If a criteria on the left is less important than the matching one on the right, put your check mark to the right of the importance 'Equal (1)' under the importance level you. Reciprocal value means the opposite effect of the factor of assigned value. Here, total eleven groups are available. Please put your mark for each group.

A. With respect to the criteria "Security Durability"

Question 1: How important is the relation between "Dependability (C1)" and "Trustworthiness (C2)"?
Question 2: How important is the relation between "Dependability (C1)" and "Human Trust (C3)"?
Question 3: How important is the relation between "Trustworthiness (C2)" and "Human Trust (C3)"?

Importance of One Criteria Over Another																		
Q.N.		9	8	7	6	5	4	3	2	1	2^{-1}	3^{-1}	4^{-1}	5^{-1}	6^{-1}	7^{-1}	8^{-1}	9^{-1}	
1	C1													✓					C2
2	C1										✓								C3
3	C2									✓									C3

B. With respect to the criteria "Dependability (C1)"

Question 1: How important is the relation between "Availability (C11)" and "Reliability (C12)"?
Question 2: How important is the relation between "Availability (C11)" and "Maintainability (C13)"?
Question 3: How important is the relation between "Availability (C11)" and "Confidentiality (C14)"?
Question 4: How important is the relation between "Availability (C11)" and "Authentication (C15)"?
Question 5: How important is the relation between "Reliability (C12)" and "Maintainability (C13)"?
Question 6: How important is the relation between "Reliability (C12)" and "Confidentiality C14)"?
Question 7: How important is the relation between "Reliability (C12)" and "Authentication (C15)"?
Question 8: How important is the relation between "Maintainability (C13)" and "Confidentiality C14)"?
Question 9: How important is the relation between "Maintainability (C13)" and "Authentication (C15)"?
Question 10: How important is the relation between "Confidentiality C14)" and "Authentication (C15)"?

Importance of One Criteria Over Another																		
Q.N.		**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**2^{-1}**	**3^{-1}**	**4^{-1}**	**5^{-1}**	**6^{-1}**	**7^{-1}**	**8^{-1}**	**9^{-1}**	
1	C11					✓													C12
2	C11									✓									C13
3	C11								✓										C14
4	C11									✓									C15
5	C12										✓								C13
6	C12										✓								C14
7	C12																	✓	C15
8	C13									✓									C14
9	C13																	✓	C15
10	C14																	✓	C15

C. With respect to the criteria "Trustworthiness (C2)"

Question 1: How important is the relation between "Availability (C21)" and "Reliability (C22)"?
Question 2: How important is the relation between "Availability (C21)" and "Maintainability (C23)"?
Question 3: How important is the relation between "Availability (C21)" and "Accountability (C24)"?
Question 4: How important is the relation between "Availability (C21)" and "Survivability (C25)"?
Question 5: How important is the relation between "Reliability (C22)" and "Maintainability (C23)"?
Question 6: How important is the relation between "Reliability (C22)" and "Accountability (C24)"?
Question 7: How important is the relation between "Reliability (C22)" and "Survivability (C25)"?
Question 8: How important is the relation between "Maintainability (C23)" and "Accountability (C24)"?
Question 9: How important is the relation between "Maintainability (C23)" and "Survivability (C25)"?
Question 10: How important is the relation between "Accountability (C24)" and "Survivability (C25)"?

Importance of One Criteria Over Another																		
Q.N.		**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**2^{-1}**	**3^{-1}**	**4^{-1}**	**5^{-1}**	**6^{-1}**	**7^{-1}**	**8^{-1}**	**9^{-1}**	
1	C21													✓					C22
2	C21										✓								C23
3	C21												✓						C24
4	C21										✓								C25
5	C22					✓													C23
6	C22										✓								C24
7	C22								✓										C25
8	C23										✓								C24
9	C23										✓								C25
10	C24									✓									C25

D. With respect to the criteria "Human Trust (C3)"

Question 1: How important is the relation between "Reliability (C31)" and "Consumer Integrity (C32)"?
Question 2: How important is the relation between "Reliability (C31)" and "Accountability (C33)"?
Question 3: How important is the relation between "Reliability (C31)" and "Confidentiality (C34)"?
Question 4: How important is the relation between "Reliability (C31)" and "Authentication (C35)"?
Question 5: How important is the relation between "Consumer Integrity (C32)" and "Accountability (C33)"?
Question 6: How important is the relation between "Consumer Integrity (C32)" and "Confidentiality (C34)"?
Question 7: How important is the relation between "Consumer Integrity (C32)" and "Authentication"?
Question 8: How important is the relation between "Accountability (C33)" and "Confidentiality (C34)"?
Question 9: How important is the relation between "Accountability (C33)" and "Authentication (C35)"?
Question 10: How important is the relation between "Confidentiality (C34)" and "Authentication (C35)"?

		Importance of One Criteria Over Another																
Q.N.		**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	2^{-1}	3^{-1}	4^{-1}	5^{-1}	6^{-1}	7^{-1}	8^{-1}	9^{-1}	
1	C31											✓							C32
2	C31									✓									C33
3	C31											✓							C34
4	C31												✓						C35
5	C32								✓										C33
6	C32											✓							C34
7	C32										✓								C35
8	C33											✓							C34
9	C33												✓						C35
10	C34								✓										C35

E. With respect to the criteria "Availability (C11)"

Question 1: How important is the relation between "Auditability (C111)" and "Feasibility (C112)"?
Question 2: How important is the relation between "Auditability (C111)" and "Accessibility (C113)"?
Question 3: How important is the relation between "Auditability (C111)" and "Software Effectiveness Evaluation (C114)"?
Question 4: How important is the relation between "Auditability (C111)" and "Operational Controls (C115)"?
Question 5: How important is the relation between "Feasibility (C112)" and "Accessibility (C113)"?
Question 6: How important is the relation between "Feasibility (C112)" and "Software Effectiveness Evaluation (C114)"?
Question 7: How important is the relation between "Feasibility (C112)" and "Operational Controls (C115)"?
Question 8: How important is the relation between "Accessibility (C113)" and "Software Effectiveness Evaluation (C114)"?
Question 9: How important is the relation between "Accessibility (C113)" and "Operational Controls (C115)"?
Question 10: How important is the relation between "Software Effectiveness Evaluation (C114)" and "Operational Controls (C115)"?

Importance of One Criteria Over Another																		
Q.N.		9	8	7	6	5	4	3	2	1	2^{-1}	3^{-1}	4^{-1}	5^{-1}	6^{-1}	7^{-1}	8^{-1}	9^{-1}	
1	C111					✓													C112
2	C111					✓													C113
3	C111								✓										C114
4	C111							✓											C115
5	C112										✓								C113
6	C112												✓						C114
7	C112										✓								C115
8	C113															✓			C114
9	C113										✓								C115
10	C114									✓									C115

F. With respect to the criteria "Reliability (C12)"

Question 1: How important is the relation between "Feasibility (C121)" and "Time-efficiency (C122)"?
Question 2: How important is the relation between "Feasibility (C121)" and "User Satisfaction (C123)"?
Question 3: How important is the relation between "Feasibility (C121)" and "Business Continuity (C124)"?
Question 4: How important is the relation between "Time-efficiency (C122)" and "User Satisfaction (C123)"?
Question 5: How important is the relation between "Time-efficiency (C122)" and "Business Continuity (C124)"?
Question 6: How important is the relation between "User Satisfaction (C123)" and "Business Continuity (C124)"?

Importance of One Criteria Over Another																		
Q.N.		9	8	7	6	5	4	3	2	1	2^{-1}	3^{-1}	4^{-1}	5^{-1}	6^{-1}	7^{-1}	8^{-1}	9^{-1}	
1	C121											✓							C122
2	C121									✓									C123
3	C121													✓					C124
4	C122								✓										C123
5	C122											✓							C124
6	C123												✓						C124

G. With respect to the criteria "Maintainability (C13)"

Question 1: How important is the relation between "Auditability (C131)" and "Scalability (C132)"?
Question 2: How important is the relation between "Auditability (C131)" and "Traceability (C133)"?
Question 3: How important is the relation between "Auditability (C131)" and "Detectability (C134)"?
Question 4: How important is the relation between "Auditability (C131)" and "Extensibility (C135)"?
Question 5: How important is the relation between "Auditability (C131)" and "Flexibility (C136)"?
Question 6: How important is the relation between "Auditability (C131)" and "Accessibility (C137)"?
Question 7: How important is the relation between "Auditability (C131)" and "Time-efficiency (C138)"?
Question 8: How important is the relation between "Scalability (C132)" and "Traceability (C133)"?
Question 9: How important is the relation between "Scalability (C132)" and "Detectability (C134)"?
Question 10: How important is the relation between "Scalability (C132)" and "Extensibility (C135)"?
Question 11: How important is the relation between "Scalability (C132)" and "Flexibility (C136)"?
Question 12: How important is the relation between "Scalability (C132)" and "Accessibility (C137)"?
Question 13: How important is the relation between "Scalability (C132)" and "Time-efficiency (C138)"?
Question 14: How important is the relation between "Traceability (C133)" and "Detectability (C134)"?
Question 15: How important is the relation between "Traceability (C133)" and "Extensibility (C135)"?
Question 16: How important is the relation between "Traceability (C133)" and "Flexibility (C136)"?
Question 17: How important is the relation between "Traceability (C133)" and "Accessibility (C137)"?
Question 18: How important is the relation between "Traceability (C133)" and "Time-efficiency (C138)"?
Question 19: How important is the relation between "Detectability (C134)" and "Extensibility (C135)"?
Question 20:How important is the relation between "Detectability (C134)" and "Flexibility (C136)"?

Question 21: How important is the relation between "Detectability (C134)" and "Accessibility (C137)"?
Question 22: How important is the relation between "Detectability (C134)" and "Time-efficiency (C138)"?
Question 23: How important is the relation between "Extensibility" and "Flexibility (C136)"?
Question 24: How important is the relation between "Extensibility (C135)" and "Accessibility (C137)"?
Question 25: How important is the relation between "Extensibility (C135)" and "Time-efficiency (C138)"?
Question 26: How important is the relation between "Flexibility (C136)" and "Accessibility (C137)"?
Question 27: How important is the relation between "Flexibility (C136)" and "Time-efficiency (C138)"?
Question 28: How important is the relation between "Accessibility (C137)" and "Time-efficiency (C138)"?

Importance of One Criteria Over Another																		
Q.N.		**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**2^{-1}**	**3^{-1}**	**4^{-1}**	**5^{-1}**	**6^{-1}**	**7^{-1}**	**8^{-1}**	**9^{-1}**	
1	C131									✓									C132
2	C131										✓								C133
3	C131										✓								C134
4	C131												✓						C135
5	C131										✓								C136
6	C131														✓				C137
7	C131											✓							C138
8	C132									✓									C133
9	C132										✓								C134
10	C132											✓							C135
11	C132																	✓	C136
12	C132										✓								C137
13	C132														✓				C138
14	C133									✓									C134
15	C133										✓								C135
16	C133									✓									C136
17	C133							✓											C137
18	C133												✓						C138
19	C134										✓								C135
20	C134										✓								C136
21	C134								✓										C137
22	C134										✓								C138
23	C135										✓								C136
24	C135									✓									C137
25	C135											✓							C138
26	C136						✓												C137
27	C136									✓									C138
28	C137												✓						C138

H. With respect to the criteria "Confidentiality (C14)"

Question 1: How important is the relation between"User Satisfaction (C141)" and "Software Effectiveness Evaluation (C142)"?
Question 2: How important is the relation between"User Satisfaction (C141)" and 'Operational Controls (C143)"?
Question 3: How important is the relation between"Software Effectiveness Evaluation (C142)" and 'Operational Controls (C143)"?

Importance of One Criteria Over Another																		
Q.N.		9	8	7	6	5	4	3	2	1	2^{-1}	3^{-1}	4^{-1}	5^{-1}	6^{-1}	7^{-1}	8^{-1}	9^{-1}	
1	C141							✓											C142
2	C141								✓										C143
3	C142										✓								C143

I. With respect to the criteria "Authentication (C15)"

Question 1: How important is the relation between "Psychological Acceptability (C151)" and "User Satisfaction (C152)"?

Question 2: How important is the relation between "Psychological Acceptability (C151)" and "Software Effectiveness Evaluation (C153)"?

Question 3: How important is the relation between "Psychological Acceptability (C151)" and "Operational Controls (C154)"?

Question 4: How important is the relation between "User Satisfaction (C152)" and "Software Effectiveness Evaluation (C153)"?

Question 5: How important is the relation between "User Satisfaction (C152)" and "Operational Controls (C154)"?

Question 6: How important is the relation between "Software Effectiveness Evaluation (C153)" and "Operational Controls (C154)"?

Importance of One Criteria Over Another																		
Q.N.		9	8	7	6	5	4	3	2	1	2^{-1}	3^{-1}	4^{-1}	5^{-1}	6^{-1}	7^{-1}	8^{-1}	9^{-1}	
1	C151											✓							C152
2	C151									✓									C153
3	C151													✓					C154
4	C152								✓										C153
5	C152											✓							C154
6	C153												✓						C154

J. With respect to the criteria "Survivability (C25)"

Question 1: How important is the relation between"Detectability (C251)" and "Extensibility (C252)"?

Question 2: How important is the relation between"Detectability (C251)" and 'Flexibility (C253)"?

Question 3: How important is the relation between"Extensibility (C252)" and 'Flexibility (C253)"?

Importance of One Criteria Over Another																		
Q.N.		9	8	7	6	5	4	3	2	1	2^{-1}	3^{-1}	4^{-1}	5^{-1}	6^{-1}	7^{-1}	8^{-1}	9^{-1}	
1	C141	✓																	C142
2	C141							✓											C143
3	C142												✓						C143

K. With respect to the criteria "Authentication (C32)"

Question 1: How important is the relation between "Psychological Acceptability (C321)" and "User Satisfaction (C322)"?

Question 2: How important is the relation between "Psychological Acceptability (C321)" and "Business Continuity (C323)"?

Question 3: How important is the relation between "Psychological Acceptability (C321)" and "Operational Controls (C324)"?

Question 4: How important is the relation between "User Satisfaction (C322)" and "Business Continuity (C323)"?

Question 5: How important is the relation between "User Satisfaction (C322)" and "Operational Controls (C324)"?

Question 6: How important is the relation between "Business Continuity (C323)" and "Operational Controls (C324)"?

Importance of One Criteria Over Another																		
Q.N.		9	8	7	6	5	4	3	2	1	2^{-1}	3^{-1}	4^{-1}	5^{-1}	6^{-1}	7^{-1}	8^{-1}	9^{-1}	
1	C321											✓							C322
2	C321													✓					C323
3	C321														✓				C324
4	C322							✓											C323
5	C322									✓									C324
6	C323										✓								C324

Your Comments (Please mark corrections as and where required): Please find details in E-Mail: kamna.sahu11@gmail.com

Try to add more references of software security and also inculcate security metrics

Kamna

Expert's Name and Signature: Kamna Sahu, Associate Consultant (TCS Noida)

(Please return this to: Rajeev Kumar (rs0414@gmail.com), D/O Information Technology, SIST, Babasaheb Bhimrao Ambedkar University, Lucknow, Uttar Pradesh, India)

Appendix D
Form for Rating the Two Versions of Software

Details and Description: Due to heavy cost incurred on software development and maintenance, secure software with longer service life span is in high demand. This property of software is termed as security durability. During software development, security durability of software can be improved by improving its attributes. Hence, the organizations need to identify, correlate, estimate and improve security durability attributes during software development. To demonstrate the same, the researcher has taken an old version of entrance exam software developed for Babasaheb Bhimrao Ambedkar University, Lucknow. With the help of developers, the researcher has evolved this software into a modified one. You are requested to rate both versions of the software. For accurate evaluation, the researcher has changed the name of designs randomly to Version 1 and Version 2. Further, ratings of the attributes may be helpful for researcher to evaluate security durability of software.

Your ratings for attributes will surely help to improve the methodology. So you are requested to kindly give your opinion for the given attributes. The scale for answers has been given in the Table 1. The responses are to be given in numeric form.

Table 1: Rating Scale

S. No.	Linguistic Value	Numeric Value of Ratings
1	Very Low (VL)	0.1
2	Low (L)	0.3
3	Medium (M)	0.7
4	High (H)	0.9
5	Very High (VH)	1.0

You need to rate the different attributes with the given scale. The ratings have been divided in five parts including 0.1 (lowest rating) while 1.0 (highest rating) of the attributes.

S. No.	Name of the Attributes	Ratings	
		Version 1	Version 2
1	Dependability		
2	Trustworthiness		
3	Human Trust		
4	Reliability		
5	Availability		
6	Authentication		
7	Maintainability		
8	Confidentiality		
9	Accountability		
10	Consumer Integrity		
11	Survivability		
12	Software Effectiveness Evaluation		
13	User Satisfaction		
14	Feasibility		
15	Operational Controls		

16	Time-efficiency		
17	Auditability		
18	Psychological Acceptability		
19	Business Continuity		
20	Accessibility		
21	Extensibility		
22	Flexibility		
23	Detectability		
24	Scalability		
25	Traceability		

Your Comments (Please mark corrections as and where required): Please find details in E-Mail:

Expert's Name and Signature:

(Please return this to: Rajeev Kumar (rs0414@gmail.com), D/O Information Technology, SIST, Babasaheb Bhimrao Ambedkar University, Lucknow, Uttar Pradesh, India)

Appendix E

Sample: Ratings Reports

Form for Rating the Two Versions of Software

Details and Description: Due to heavy cost incurred on software development and maintenance, secure software with longer service life span is in high demand. This property of software is termed as security durability. During software development, security durability of software can be improved by improving its attributes. Hence, the organizations need to identify, correlate, estimate and improve security durability attributes during software development. To demonstrate the same, the researcher has taken an old version of entrance exam software developed for Babasaheb Bhimrao Ambedkar University, Lucknow. With the help of developers, the researcher has evolved this software into a modified one. You are requested to rate both versions of the software. For accurate evaluation, the researcher has changed the name of designs randomly to Version 1 and Version 2. Further, ratings of the attributes may be helpful for researcher to evaluate security durability of software.

Your ratings for attributes will surely help to improve the methodology. So you are requested to kindly give your opinion for the given attributes. The scale for answers has been given in the Table 1. The responses are to be given in numeric form.

Table 1: Rating Scale

S. No.	Linguistic Value	Numeric Value of Ratings
1	Very Low (VL)	0.1
2	Low (L)	0.3
3	Medium (M)	0.7
4	High (H)	0.9
5	Very High (VH)	1.0

You need to rate the different attributes with the given scale. The ratings have been divided in five parts including 0.1 (lowest rating) while 1.0 (highest rating) of the attributes.

S. No.	Name of the Attributes	Ratings	
		Version 1	Version 2
1	Dependability	VH	H
2	Trustworthiness	H	H
3	Human Trust	L	H
4	Reliability	H	H
5	Availability	VL	VH
6	Authentication	VL	M
7	Maintainability	M	H
8	Confidentiality	L	H
9	Accountability	VH	H
10	Consumer Integrity	M	H
11	Survivability	H	M
12	Software Effectiveness Evaluation	H	H
13	User Satisfaction	H	H
14	Feasibility	L	M

15	Operational Controls	VH	H
16	Time-efficiency	VL	H
17	Auditability	M	M
18	Psychological Acceptability	H	H
19	Business Continuity	M	M
20	Accessibility	VH	H
21	Extensibility	H	H
22	Flexibility	VL	L
23	Detectability	M	M
24	Scalability	M	M
25	Traceability	M	M

Your Comments (Please mark corrections as and where required): Please find details in E-Mail:

—

Expert's Name and Signature: Chanchal, SAQ, Infosys, Lucknow.

(Please return this to: Rajeev Kumar (rs0414@gmail.com), D/O Information Technology, SIST, Babasaheb Bhimrao Ambedkar University, Lucknow, Uttar Pradesh, India)

Appendix F
Certificate from Software Industry

SAQ Infosys

Dated: 15/11/2018 **Ref.No.007**

To Whomsoever It may Concern

This is to certify that **Mr. Rajeev Kumar** PhD Scholar from Department of Information Technology, Babaseahab Bhimrao Ambedkar University (A Central University), Lucknow, U.P., India has conducted his study on the two successive versions of Online Entrance Test Module upgraded for Babasaheb Bhimrao Ambedkar University. He has used the project details along with other details for research purpose. Our team members have adopted his suggestions for improving secure service life of software.

The identification of the thesis work conducted as per our desire and organization's policy. The data that is used in the thesis is correct to the best of my knowledge and belief.

Authorized Signatory

Sector- 4/250 Vikas Nagar Lucknow - 226022 Cell–09335611550 E-mail: saqinfosys@gmail.com

Appendix G
Plagiarism Report

URKUND

Urkund Analysis Result

Analysed Document:	Thesis Complete without figures.doc (D43086962)
Submitted:	10/26/2018 8:27:00 AM
Submitted By:	gbl.bbau@gmail.com
Significance:	6 %

Sources included in the report:

Nilu Singh.docx (D25224778)
Alka mam Fuzzy-Delphi Analytical Hierarchy Process for Importance of Security Attributes.docx (D29687514)
https://hal.archives-ouvertes.fr/tel-01074958v1
https://link.springer.com/chapter/10.1007/978-981-10-2750-5_49
https://www.researchgate.net/profile/Rajeev_Kumar123/publication/283055165_Revisiting_Software_Security_Durability_Perspective/links/5627f1ca08ae04c2aead80f6.pdf?origin=publication_list

Instances where selected sources appear:

85